Visual Data Analysis Supported by Eye-Tracking, Multi-Touch Displays, and Machine Learning

Mohammad Chegini, MSc

Visual Data Analysis Supported by Eye-Tracking, Multi-Touch Displays, and Machine Learning

Mohammad Chegini, MSc

Doctoral Thesis

to achieve the university degree of

Doctor of Philosophy

PhD degree programme: Computer Science

submitted to

Graz University of Technology
Nanyang Technological University

Supervisor

Univ.-Prof. Dr.rer.nat. M.Sc. Tobias Schreck
Institute of Computer Graphics and Knowledge Visualisation (CGV), Graz University of Technology

Co-Supervisors

Assoc.-Prof. Dr. M.Eng. Alexei Sourin
School of Computer Science and Engineering, Nanyang Technological University

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Keith Andrews
Institute of Interactive Systems and Data Science (ISDS), Graz University of Technology

26 Nov 2020

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

10.03.2021

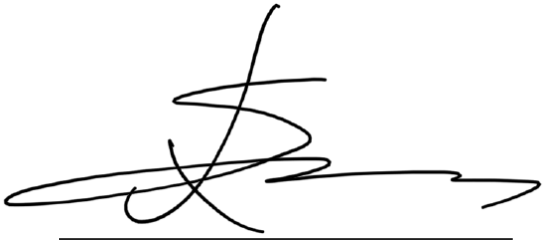
Date

Mohammad Chegini

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

8.03.2021



Date

Prof. Alexei Sourin

Authorship Attribution Statement

This thesis contains material from nine papers published in the peer-reviewed journals and conferences. The list of publications is in Section 8.2. The name of contributors and their contributions are listed in Section 1.4.

10.03.2021

Date

Chegini

Mohammad Chegini

Abstract

In recent years, data analysts have been confronted by increasing amounts of data, often in the form of multivariate datasets. Multivariate datasets can be thought of as a table, where dimensions are columns, and records are rows. Machine learning and data mining algorithms can help an analyst to build machine learning (ML) models to find structures in a dataset algorithmically. Alternatively, visualisation techniques such as scatterplot, scatterplot matrix, and parallel coordinates can help an analyst explore and find structures in a dataset visually. Although extensive research has been done around building and visualising an ML model, there is less research linking ML models and visualisations through human-centred interactions. Such a connection has the potential to help an analyst build better ML models by interactively steering the process. However, designing and evaluating such interaction techniques is challenging.

In this thesis, visual analytics techniques are proposed, which focus on building and modifying an ML model of a multivariate dataset, using machine learning, visualisation, and interactions. Moreover, the use of novel interaction modalities and devices such as large multi-touch displays, handheld devices, and eye-trackers is explored.

As a first step, a novel approach for selecting, searching for, and comparing local patterns within multivariate datasets using scatterplots is presented. An analyst can select a part of a scatterplot from a scatterplot matrix, and search for similar patterns using both model-based (ML regression) descriptors and shape-based descriptors. A relevance feedback module enables the analyst to improve the regression analysis and find relevant patterns more effectively.

The second part of the thesis goes beyond simple interaction and exploration using an ML model and focuses on ML model creation and modification. Specifically, an interactive visual labelling technique is presented, which allows an analyst to build and interactively improve an (ML classification) model for multivariate datasets. The technique combines linked visualisations, clustering, and active learning to help an analyst interactively label a multivariate dataset. In the third step, a user study was conducted which showed that such an interactive labelling technique could surpass common active learning algorithms for building an effective ML model.

Finally, the fourth part of the thesis explores several novel interaction modalities. It is shown how large multi-touch displays are effective for collaborative analysis of scatterplots. Extending these interactions, analysts can use a secondary handheld device to interact with linked-view information visualisation application to label multivariate datasets. In addition, user eye gaze interaction can be garnered by the system to help re-arrange the axes in a parallel coordinates visualisation.

In summary, this thesis uses human-centred interactions to bridge the gap between ML techniques and visualisation techniques. The thesis presents how to (1) interactively search and explore local regression models in a scatterplot space, (2) interactively build and improve an ML model of a multivariate dataset by linked visualisations, clustering, and active learning, and (3) use eye-tracking and multi-touch displays to investigate regression ML models collaboratively, and use eye gaze as an input for interaction with visualisations of a multivariate dataset.

Contents

List of Figures

List of Tables

Acknowledgements

Even though one person shall carry the title, doing a PhD is a team effort. I would like to start by thanking my supervisors. I want to thank Prof. Tobias Schreck for giving me the opportunity to come to Graz, and for showing compassion and helping me both in my professional and personal life. I want to thank Prof. Keith Andrews for forwarding my application to Tobias, and supporting me on my research and life goals in Graz. I want to thank Prof. Alexei Sourin for all his help in make me a better researcher. Without his help, I would not be able to continue my research in Singapore. I would like to thank Prof. Ali Asghar Nazari Shirehjini for teaching me the basics of research before starting my PhD. I would like to thank Prof. Ebrahim Abtahi, Prof. Sadegh Aliakbari, Prof. Omid Fatemi, and Prof. Mansour Jamzad for everything they have done for me during my studies.

I thank Prof. Heidrun Schumann and Christian Tominski for hosting me in Rostock and giving me new directions for research. I especially thank Prof. Jürgen Bernard for hosting me in Darmstadt and changing the direction of my thesis towards a better one. Many thanks to Prof. Olga Sourina and Prof. Wolfgang Müller-Wittig for their extraordinary hospitality during my stay at Fraunhofer Singapore.

I want to thank all my co-authors of papers I wrote during my doctoral degree: Lin Shao, Dirk J Lehmann, Keith Andrews, Tobias Schreck, Philip Berger, Heidrun Schumann, Christian Tominski, Robert Gregor, Alexei Sourin, Jian Cui, Fatemeh Chegini, and Jürgen Bernard.

Being an expat is hard, I was lucky to feel at home during my stays in Graz and Singapore, thanks to my colleagues. I would like to thank all my colleagues at CGV, Fraunhofer Singapore, and Fraunhofer Austria. I would like to thank Ingrid Preininger, Heike Graf-Gürtler, Prof. Ursula Augsdörfer, Lars Schimmer, Wolfgang Scheicher, and Ulrich Krispel sincerely for helping me beyond the professional level during my stay in Graz. I would like to thank Jiaen Koh for being there for me. I thank my office mates Madeley Karina Coaquira Congona, Nicole Weidinger, Xingzi Zhang, and Jian Cui for making my life easier. I wrote most of my thesis during COVID-19 pandemic, I thank my flatmate and friend Lukas Freier for letting me turn the kitchen into the home office. I thank all my friends that made this journey awesome.

Looking back, my journey started by leaving home in Tehran in January 2017. I want to thank my three siblings, Narges, Zeinab, and Fatemeh for showing me the way. I knew the basics of research from a young age when my father was laying his research papers on the ground, and I was looking through them. I recall that the mathematics in those papers seemed terrifying then, and I stand by it even today. I thank my father for his support during all these years.

And in 2002, there was this lady in her late thirties that was teaching a twelve year old boy how to draw a circle on the blue screen using QBasic. Sometimes I wonder if that magical moment paved my way into computer graphics. I would like to thank my mother for believing in me.

Mohammad Chegini
Graz, Austria, 26 Nov 2020

Chapter 1

Introduction

"The tale of me and my beloved shall not have a closure;
After all, anything without a beginning won't have an end."

[Hafez, Persian poet, 1315-1390.]

With the advent of new technologies for data gathering and storage, analysts are facing a large amount of data that cannot be processed or analysed with traditional tools. Information Visualisation (InfoVis) and Machine Learning (ML) [Samuel 1959] are becoming two pillars to empower the analyst exploring data and making crucial decisions. Despite advances in both fields, there is a gap between these two disciplines that is yet to be addressed. InfoVis relies mainly on the power of humans' brain, and ML on the computational ability of machines. Therefore, a successful combination of these two disciplines will unravel complex problems.

1.1 Information Visualisation

Visual data analysis by the help of information visualisation has become a key area in computer science and an established approach to empower domain experts. Information Visualisation is "the use of computer-supported, interactive, visual representations of abstract data to amplify cognition" [Card et al. 1999]. Keim et al. [2008] define Visual Analytics (VA) as "the science of analytical reasoning facilitated by interactive visual interfaces". VA considers approaches and systems to help data analysts explore and make sense of large, complex datasets, often in a context of decision making and to find unknown patterns. Visual analytics systems combine appropriate approaches from, among others, InfoVis, Human-Computer Interaction, Data Analysis/Data Mining, User Evaluation, and Machine Learning. The applications of VA are vast, including but not limited to multivariate data exploration on scatterplots [Shao, Mahajan et al. 2017], interactive labelling [Bernard, Zeppelzauer, Sedlmair et al. 2018], and subspace search [Tatu et al. 2012].

InfoVis techniques should be adapted according to data types. For the particular case of *multivariate data*, i.e. tabular data having a large number (n) of dimensions and (m) of data records, multiple linked views visualisation has become popular. Multiple linked views are often used to gain a better understanding of a high-dimensional dataset. Such views are usually connected by techniques such as brushing or combined navigation [Roberts 2005]. Among possible views for representing multivariate data are scatterplot, SPLOM, and parallel coordinates.

Scatterplots are bivariate projections of pairs of dimensions. For a multivariate data space of n dimensions, n^2 pairwise scatterplots are required to completely visualise the space ($n^2/2$ if transposed plots are eliminated). A matrix of scatterplots representing every pairwise combination of plots is called

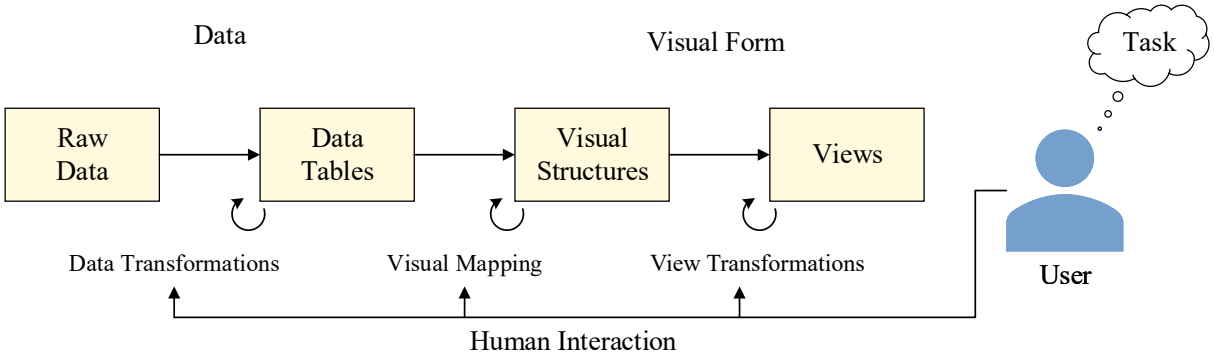


Figure 1.1: The visualisation pipeline (adapted from [Card et al. 1999]).

a SPLOM. The parallel coordinates visualisation shows the dimensions of a dataset as parallel vertical axes and its records as horizontal polylines [Inselberg 1985]. Parallel coordinates provide a concise overview of the entire dataset and are suitable for exploring correlations between neighbouring dimensions. A parallel coordinates plot has been shown to outperform individual scatterplots when the task requires interaction with more than two dimensions [Netzel et al. 2017].

Regardless of representation techniques, InfoVis is not complete without interaction. Carl et al. [Card et al. 1999] defines visualisation as the "mapping of data to a visual form that supports human interaction in a workplace for visual sense-making". Figure 1.1 shows the visualisation pipeline, which starts with raw data and continues toward views. Through interaction, the user can connect to this pipeline and explore the data.

The medium of interaction plays an essential role. InfoVis techniques should be tailored for different displays and interaction devices. The concept of interaction in InfoVis and VA has a long history [B. Lee et al. 2012]. Nevertheless, novel device and display technologies, and novel multimodal interaction devices [B. Lee et al. 2018] including gesture recognition, eye tracking, or data physicalisation offer even more possibilities for InfoVis.

1.2 Machine Learning and Visual Analytics

As the ability to extract and store data significantly increased over recent years, VA needs to rely on the computational power of machines. ML involves algorithms which improve automatically through experience. ML algorithms build a mathematical model (called an ML model) based on a training dataset of sample data records, in order to make predictions or decisions about future data records. ML techniques can be grouped into supervised, unsupervised, and reinforcement learning methods.

Supervised ML algorithms build an ML model based on a fully labelled training dataset, where both the input records and desired outcomes are fully specified in advance. The two most important supervised ML models are regression models and classification models [Nilsson 1965]. For a multivariate dataset, such an ML model can be used to predict the outcome for a new record. For example, consider a multivariate dataset consisting of age and weight as input, and height as output. With a proper ML model, the computer can predict the height of a new record, after receiving age and weight as inputs.

Unsupervised ML techniques are applied to unlabelled datasets. Clustering [Wenskovitch et al. 2018] is the prominent example of an unsupervised ML technique. Clustering techniques are useful for finding similar records within a multivariate dataset. The instances in each group will get the same label, and thus an ML model is created.

Figure 1.2 shows the visual analytics pipeline defined by Keim et al. [2008]. Models (in this thesis, ML models) can be created from data using data mining techniques. Visualisations can foster the ML

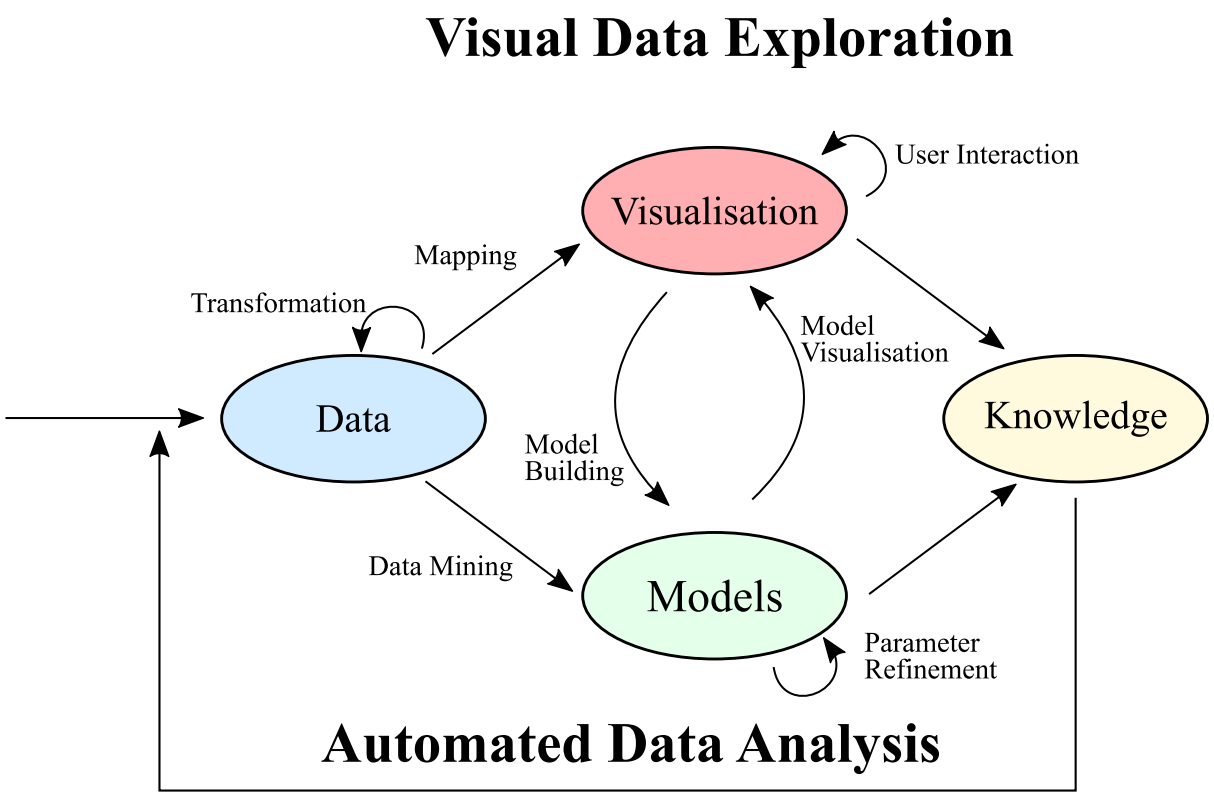


Figure 1.2: Visual analytics pipeline (adapted from [Keim et al. 2008]).

model creation, and the ML model can be visualised for further exploration.

For ML techniques, creating a model with high accuracy is the first and the most important step. Supervised algorithms use a set of labelled data, and unsupervised algorithms use a set of unlabelled data to build a model. In the traditional approach, analysts label as much data as possible to build an ML model with high accuracy. When the number of classes and the size of dataset increases, the task of labelling can become extraordinarily tedious and time-consuming. Therefore, it is important to label the data instances that have the highest impact on the accuracy of the model. Active Learning (AL) techniques help the analyst to label instances that have the highest impact on the accuracy of the model incrementally [Settles 2012].

Active Learning strategies interactively collect new labelled records by judiciously asking for additional input from the user. To make the process more effective and efficient, it is crucial for the system to propose records for interactive labelling wisely, choosing those records which are most likely to improve the underlying ML model. Figure 1.3 shows the cycle of active learning, in which the human annotator label instances that are selected by the algorithm to improve the ML model.

There are several occasions that ML algorithms can benefit from InfoVis. Quality assurance, building an ML model, and parameter optimisation are among some of them. The user can quickly check the quality of data and the ML model by visual inspection. In the case of multivariate data and multiple linked views, instances that belong to the same category are visually closer to each other. The analyst can further utilise the visual information to label instances. This process is called visual labelling. Visual labelling is specifically useful for building an ML model with a limited number of instances. InfoVis can help the analyst to tune parameters of an ML algorithm. While parameters of ML algorithms can be automatically defined, the user should be able to control them interactively.

Apart from visual inspection and parameter optimisation for ML, the focus of this thesis is on user interaction with visualisation for building and modifying the ML model. For the interaction part, both

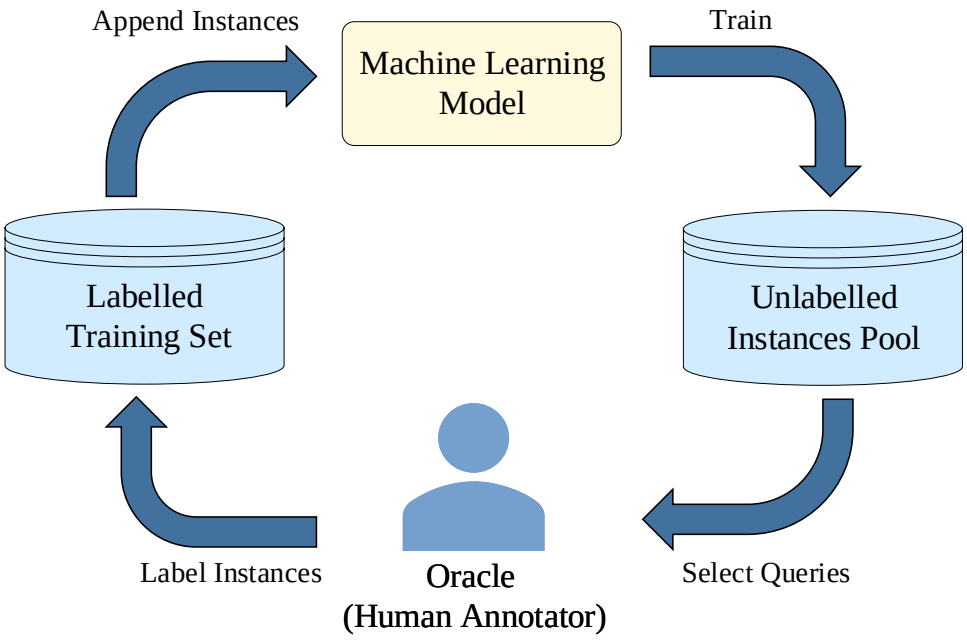


Figure 1.3: Active learning cycle (adapted from [Yu et al. 2015]).

traditional interactions such as mouse and keyboard, and novel interaction devices such as eye-trackers, and multi-touch displays are investigated. Moreover, several VA techniques are presented to build ML models more effectively in comparison to traditional automatic data mining and machine learning algorithms.

1.3 Research Questions

In this thesis, four research questions and corresponding hypotheses are formed and answered. The common goal of all questions is the role of interaction in VA for building and improving an ML model.

Research Question 1 (RQ1): *How to use VA to find structures in an ML model?*

Research Question 2 (RQ2): *How to use VA to build an ML model?*

Research Question 3 (RQ3): *How to compare VA techniques with traditional automated algorithms for building ML models?*

Research Question 4 (RQ4): *How to use non-traditional interactions to improving the building and exploration of an ML model and to foster collaboration in teams?*

Analysts face a tremendous amount of data, especially when it comes to multivariate datasets. To explore these datasets, they often use scatterplots. Although scatterplots are effective for visualising multivariate datasets [Sarikaya and Gleicher 2018], further interaction techniques are needed to find and visualise structures in the ML model created from the dataset. Therefore, RQ1 was formulated to investigate the suitability of common multivariate data visualisation techniques for finding structures in ML models. Specifically, in Chapter 3, a pipeline for finding local patterns in a SPLOM using regression models is proposed.

To go beyond finding structures in a finished ML model, RQ2 looks at VA techniques which can support an analyst to build and improve the ML model interactively. Chapters 4 and 5 consider classification models and show how a combination of VA and ML can surpass active learning algorithms for labelling

a multivariate dataset (RQ3).

Finally, Chapter 6 addresses RQ4. Firstly, it is shown how large multi-touch displays and regression analysis can be used for collaborative analysis of scatterplots. Secondly, by linking a handheld device to large displays, the collaboration concept is extended from two analysts to a small group. The last part of Chapter 6, explores the use of gaze information for ordering axes in a parallel coordinates plot.

1.4 Contributions

The main contributions of this thesis are described in Chapters 3, 4, 5, and 6.

1.4.1 Local Scatterplot Patterns (Chapter 3)

Analysts often use visualisation techniques like a scatterplot matrix (SPLOM) to explore multivariate datasets. The scatterplots of a SPLOM can help to identify and compare two-dimensional global patterns. However, *local* patterns, which might only exist within subsets of records, are typically much harder to identify and may go unnoticed among larger sets of plots in a SPLOM. Chapter 3 explores the notion of local patterns and presents a novel approach to visually select, search for, and compare local patterns in a multivariate dataset. Regression models are used to define model-based descriptors for local regions in scatterplots. Together with shape-based pattern descriptors, these are used to automatically compare local regions in scatterplots and assist in the discovery of similar local patterns. Mechanisms are provided to assess the level of similarity between local patterns and to rank similar patterns effectively. Moreover, a relevance feedback module is used to suggest potentially relevant local patterns to the user. The approach has been implemented in an interactive tool and demonstrated with two real-world datasets and use cases. It supports the discovery of potentially useful information such as clusters, functional dependencies between variables, and statistical relationships in subsets of data records and dimensions.

The work presented in Chapter 3 is based on [Chegini, Shao, Gregor et al. 2018].

Contributors In the aforementioned article, Lin Shao contributed to the related work section. Robert Gregor helped with the formulation of the recommender system. Dirk Lehmann prepared a draft for the introduction section. Dirk Lehmann, Keith Andrews, and Tobias Schreck contributed to the definition of underlying research questions and revised and finalised the manuscript.

1.4.2 Interactive Visual Labelling of Multivariate Datasets (Chapter 4)

Supervised machine learning techniques require labelled multivariate training datasets. Many approaches address the issue of unlabelled datasets by tightly coupling machine learning algorithms with interactive visualisations. Using appropriate techniques, analysts can play an active role in a highly interactive and iterative machine learning process to label the dataset and create meaningful partitions. While this principle has been implemented either for unsupervised, semi-supervised, or supervised machine learning tasks, the combination of all three methodologies remains challenging.

In Chapter 4, a visual analytics approach is presented which combines a variety of machine learning capabilities for building a classification model with four linked visualisation views, all integrated within the mVis (**m**ultivariate **Vis**ualiser) system. The available palette of techniques allows an analyst to perform exploratory data analysis on a multivariate dataset and divide it into meaningful labelled partitions, from which a classifier can be built. In the workflow, the analyst can label interesting patterns or outliers in a semi-supervised process supported by active learning. Once a dataset has been interactively labelled, the analyst can continue the workflow with supervised machine learning to assess to what degree the subsequent classifier has effectively learned the concepts expressed in the labelled training dataset. Using a novel technique called automatic dimension selection, interactions the analyst had with dimensions of the multivariate dataset are used to steer the machine learning algorithms.

A real-world football dataset is used to show the utility of mVis for a series of analysis and labelling tasks, from initial labelling through iterations of data exploration, clustering, classification, and active learning to refine the named partitions, to finally producing a high-quality labelled training dataset suitable for training a classifier. The tool empowers the analyst with interactive visualisations including scatterplots, parallel coordinates, similarity maps for records, and a new similarity map for partitions.

The work presented in Chapter 4 is based on [Chegini, Bernard, Berger et al. 2019], and [Chegini, Bernard, Shao et al. 2019].

Contributors In the aforementioned publications, Jürgen Bernard wrote the draft of the introduction section. Philip Berger helped to shape and write the use case study. Jürgen Bernard, Keith Andrews, and Tobias Schreck contributed to the definition of underlying research questions. Keith Andrews, Tobias Schreck, and Alexei Sourin revised and finalised the manuscript.

1.4.3 Active Learning Versus Interactive Labelling (Chapter 5)

Methods from supervised machine learning allow the classification of new data automatically and are tremendously helpful for data analysis. The quality of supervised learning depends not only on the type of algorithm used but, importantly, also on the quality of the labelled dataset used to train the classifier. Labelling instances in a training dataset is often done manually, relying on selections and annotations by expert analysts, and is often a tedious and time-consuming process.

Active learning algorithms can automatically determine a subset of data instances for which labels would provide useful input to the learning process. Interactive visual labelling techniques are a promising alternative, providing effective visual overviews from which an analyst can simultaneously explore data records and select items to a label. By putting the analyst in the loop, higher accuracy can be achieved in the resulting classifier. While initial results of interactive visual labelling techniques are promising in the sense that user labelling can improve supervised learning, many aspects of these techniques are still largely unexplored.

Chapter 5 presents a study conducted using the mVis tool to compare three interactive visualisations (similarity map, SPLOM with scatterplot, and parallel coordinates) with each other and with active learning for the purpose of labelling a multivariate dataset. The results show that all three interactive visual labelling techniques surpass active learning algorithms in terms of classifier accuracy and that users subjectively prefer the similarity map over SPLOM with scatterplot and parallel coordinates for labelling. Furthermore, users employed different labelling strategies depending on the visualisation being used.

The work presented in Chapter 5 is based on [Chegini et al. 2020].

Contributors In the aforementioned article, Jürgen Bernard designed the study. Fatemeh Chegini drafted the manuscript. Jian Cui helped to conduct the experiment and data processing. Alexei Sourin, Keith Andrews, and Tobias Schreck contributed to the definition of the underlying research questions, and revised and finalised the manuscript.

1.4.4 Multimodal Interaction for Data Analysis (Chapter 6)

Current advances in human-computer interaction introduce novel modalities such as eye-gaze, speech, and multi-touch interfaces. These input modalities bring new opportunities to design visual analytics techniques for multivariate data interaction. On the other hand, machine learning techniques can help to perform tasks that are currently done manually by analysts. Combining new interaction methods with state-of-the-art machine learning algorithms brings challenges that are yet to be solved. The focus of Chapter 6 is on using novel interaction modalities such as multi-touch interfaces and eye-gaze together with machine learning algorithms to improve the exploration of patterns, especially in scatterplots. This combination creates interactive visual systems and allows the analyst to explore multivariate datasets

more effectively. Chapter 6 is based on [Chegini et al. 2017], [Chegini, Shao, K. Andrews and Schreck 2018], [Chegini, K. Andrews et al. 2019b], and [Chegini, K. Andrews et al. 2019a].

Contributors In [Chegini et al. 2017], Lin Shao provided the regression lens technique that later was further developed for the paper. Dirk Lehmann wrote the draft ot the introduction. Dirk Lehmann, Keith Andrews, and Tobias Schreck contributed to the definition of the underlying research questions, and revised and finalised the manuscript. Alexei Sourin, Keith Andrews, and Tobias Schreck had the same role in [Chegini, Shao, K. Andrews and Schreck 2018], [Chegini, K. Andrews et al. 2019b], and [Chegini, K. Andrews et al. 2019a].

1.5 List of Abbreviations

Table 1.1 shows common terms that are used in the thesis together with their abbreviations.

Abbreviation	Term
mVis	**m**ultivariate **Vis**ualiser
InfoVis	Information Visualisation
VA	Visual Analytics
SPLOM	Scatterplot Matrix
IVL	Interactive Visual Labelling
AL	Active Learning
ML	Machine Learning

Table 1.1: Abbreviations used in the thesis.

Chapter 2

Related Work

"Being wise won't turn the wheel of cosmos in your favour."

[Khayyam, Persian mathematician and poet, 1048–1131.]

The first step for the research is to study related work and find the gap between state of the art technologies with real-world problems. This chapter gives an overview of visual data analysis, and its application for scatterplot exploration, interactive visual labelling, and multimodal interaction.

2.1 Scatterplot Exploration

The approach described in Chapter 3 is related to several research areas of scatterplot exploration. The three most important directions which influenced the work in Chapter 3 are local scatterplot segmentation, visual retrieval techniques, and visualisation of scatterplot patterns.

2.1.1 Scatterplot Segmentation

In recent years, the segmentation of local patterns has become an accepted and even essential part of data analysis in many fields, including genome research [Eisen et al. 1998], trajectory analysis [Mann et al. 2002], and image processing [Friedman and Russell 1997]. A pattern can be defined as a set of records in a scatterplot that is contained inside a bounding box. A global pattern consists of all records in a scatterplot, whereas a local pattern consists of a subset of records.

Data scientists often use data mining techniques in combination with information visualisation to present extracted patterns visually for human perception. Mayorga and Gleicher [2013] describe an automatic abstraction approach which groups dense data points to reveal the relationship between data subgroups. Shao et al. [2016] extracted local scatterplot motifs to create a visual overview of frequent patterns, which was then used to rank scatterplot views based on an adapted TF-IDF algorithm from information retrieval. The idea was to automatically determine weights of interest for a pattern, by comparing their occurrence frequencies within and among scatterplots. Another approach uses sensitivity coefficients from flow field analysis to highlight the local variation of one variable in relation to another [Chan et al. 2010]. Chen et al. [2014] used a hierarchical multi-class sampling technique to create new visual abstraction schemes for scatterplot visualisations. Sedlmair, Tatu et al. [2012] proposed a taxonomy of visual cluster separation factors in scatterplots and a data-driven framework for evaluating visual quality measures [Sedlmair et al. 2015].

The technique described in Chapter 3 uses a sliding-window approach for segmentation, which compares a query against many possible matching candidate positions and areas, hence implicitly and heuristically segmenting the data.

2.1.2 Search Techniques for Scatterplot Retrieval

Another essential part of the work in this thesis is the description and retrieval of scatterplot patterns. This research topic has been extensively investigated and addressed in recent works. Some of the pioneering work includes the Scagnostics approach by Wilkinson et al. [2005], which characterises 2D point distributions in a multidimensional Euclidean space using graph-theoretic measures. The approach can be used to search for patterns based on density, skewness, shape, outliers, and texture [Matute et al. 2018]. Similar approaches were developed with the aim of finding similar patterns in other application domains such as time series or image retrieval [Nhon et al. 2013; Nhon and Wilkinson 2014].

Scherer et al. [2011] introduced a goodness-of-fit approach based on regression models to find functional dependencies between pairs of variables in a dataset. To search for patterns of interest, the user can either enter the query directly as a formula or sketch a scatterplot. Scherer et al. [2013] extended the approach to compare sets of scatterplots based on a bag-of-words model derived from scatterplot descriptors. Scherer et al. [2012] compared scatterplot descriptors for effectiveness in finding globally similar scatterplots based on a defined ground truth dataset. Shao et al. [2014] considered image-based features for sketch-based search in scatterplot data, including real-time feature extraction of the sketch.

Interesting work also exists to automatically detect a pattern in higher-dimensional data spaces. For instance, Tatu et al. [2012] introduced a subspace search algorithm which suggests a set of subspaces of interest.

2.1.3 Visualisation of Local Patterns

Various visual approaches have been proposed to present local properties in a scatterplot. Yates et al. [2014] described an enhanced SPLOM representation called Glyph SPLOM, which links heatmap properties to a SPLOM. Instead of showing all single scatterplots, it uses glyphs to visually encode similarity features based on the occupancy of the scatterplot quadrants. The Regression Lens [Shao, Mahajan et al. 2017] is an example of how local properties of a scatterplot can be displayed interactively. Users can apply an interactive regression analysis on a local portion of the data and immediately see the best fitting regression model on the plot. Eisemann et al. [2014] describe interactive visualisation of distinct patterns of data within a given scatterplot (a hierarchy of localised scatterplots), which allows the user to explore dense areas in a scatterplot. In Chapter 3, a combination of these approaches is used to enhance the visualisation of local patterns and facilitate the exploration of the dataset.

There are two essential aspects when visually analysing local patterns, namely, visualising and aggregating local patterns, and supporting visual comparison between them. Schreck and Panse [2007] used class labels to group data points in a scatterplot, and show properties of the contained points using aggregation by bounding boxes, circles or convex hulls. Also, colour or blur was used to convey properties of the groups. Tominski et al. [2012] suggested three visual comparison methods based on the natural behaviour of users when comparing charts: side-by-side, shine through, and folding interaction. Gleicher et al. [2011] argued that since comparing complex objects is difficult, a promising strategy is the abstraction of complexity. They presented three types of comparative visualisation: juxtaposition, superposition, and explicit encoding.

The work described in Chapter 3 uses all three approaches to build a novel technique for finding local patterns in scatterplots spaces.

2.1.4 Delineation of the Approach and Novelty in Chapter 3

The work described in Chapter 3 differs in that the query is selection-based instead of sketch-based and the search algorithm combines both model-based and shape-based descriptors to specifically address local patterns. Moreover, the user can select the most relevant matches to further refine the search query. In Chapter 3, it is explained how by exploring the scatterplot space using visual analytics and feedback

	Visual Clustering	Clustering	Classification	Active Learning
ML Type	Unsupervised	Unsupervised	Supervised	Semi-Supervised
Existing Labels	Not Required	Not Required	Required	Required
Records to Label	Chosen by user.	All unlabelled records.	Unlabelled records closer than a threshold to a label.	Specific number of records chosen strategically.
Creates Partitions	By User	Yes	No	No
Algorithms	PCA, MDS, t-SNE	K-means, Hierarchical	Random Forest	Random Forest
Triggered By	User	User	User	System

Table 2.1: Techniques which support interactive labelling of records.

loop, the analyst finds relationships between variables locally that are unknown to him. Without the technique described in this thesis, if the features of the dataset are unknown, the current state of the art tools cannot find these patterns within a reasonable time frame.

2.2 Interactive Visual Labelling

VA applications benefit from both unsupervised and supervised Machine Learning (ML) algorithms to support data exploration and analytical reasoning [Endert et al. 2018]. Table 2.1 gives an overview of some of the techniques which support interactive labelling. Unsupervised machine learning techniques can be applied to unlabelled datasets, since they do not require any training data. For example, clustering techniques [Wenskovitch et al. 2018] can be used to find groupings of similar records within a dataset. Exploratory information visualisations can be used to visually cluster (and then select) records according to their similarity or dissimilarity, since similar records are typically closer together in the visualisation. Semi-supervised ML techniques [Settles 2012] require at least some labelled data records before they can be used. In active learning, some labelled data records are provided, and the system interactively collects new examples through additional input from the user. Supervised ML techniques such as classification [Choo et al. 2010] require a proper training set of labelled records.

2.2.1 Visual Clustering

Exploratory information visualisations can be used as interactive interfaces to select (groups of) similar records or to identify and select outliers. Scatterplots visualise records along two chosen dimensions. Records which are similar (in those two dimensions) are plotted close together. Dimensionality reduction and projection methods can be used to generate a *similarity map*, which visually infers a clustering by spatial proximity. Records closer together in the projected similarity map are more similar to one another in the high-dimensional space [Sacha, Zhang et al. 2017]. In parallel coordinates [Inselberg 1985], similar records are represented by polylines which follow similar paths. It is also possible to filter records by ranges on each dimension.

Cluster Sculptor [Bruneau et al. 2015] is an interactive clustering system which allows the user to update the cluster labels of a dataset iteratively. The system relies on a t-SNE projection view, label diffusion, and dissimilarity transform techniques. H. Lee et al. [2012] built a system called iVisClustering based on latent Dirichlet allocation (LDA), which helps the user to perform clustering with interactive visualisation, including parallel coordinates and scatterplots. RCLens [Lin et al. 2017] supports the identification and exploration of rare categories (minority classes), utilising an active learning algorithm

to help the analyst iteratively finds rare categories within the dataset. In Chapter 4, interactive clustering is used to guide the analyst in finding some preliminary structure in the dataset.

2.2.2 Clustering

Classic clustering techniques such as k-means [Lloyd 1982] and hierarchical clustering [Karypis et al. 1999] are used to form groups (partitions) of records according to their similarity. The result of these clustering algorithms can be visually inspected. In early work, gCluto [Rasmussen and Karypis 2004] allowed an analyst to visually inspect clusters created by running multiple clustering techniques while tuning the parameters. Nam et al. [2007] proposed a technique allowing analysts to tune the parameters of clustering algorithms interactively to find suitable clusters based on the user's needs. The technique was proposed and tested on high-dimensional datasets. Later, Andrienko et al. [2009] suggested a general approach to find clusters in large sets of spatial data objects and demonstrated the approach on a dataset of trajectories. Kwon et al. [2018] developed Clustervision, which clusters a dataset with various clustering algorithms, and ranks and visualises clustering results based on quality metrics, allowing analysts to choose the most suitable for their purpose.

2.2.3 Classification

Classification is a supervised ML technique which can identify to which class a record belongs, given a sufficiently large training set of labelled records. VA can help classification algorithms by adding the knowledge of the user in an iterative manner [Paiva et al. 2015]. For example, iVisClassifier [Choo et al. 2010] supports a user-driven classification process, where the analyst explores multi-dimensional data through a supervised dimensionality reduction and performs classification.

2.2.4 Active Learning

Known active learning strategies include looking for helpful records a) near decision boundaries of margin-based classifiers [Wu et al. 2006; Tuia et al. 2011]), b) with high entropy of class probabilities [Settles and Craven 2008], c) with high uncertainty of a committee of classifiers [Seung et al. 1992; Mamitsuka 1998], or d) to reduce risk [Qi et al. 2009] or variance [Hoi et al. 2006].

Common active learning strategies include Smallest Margin [Scheffer et al. 2001; Wu et al. 2006], Entropy-Based Sampling [Settles and Craven 2008], and Least Significant Confidence [Culotta and McCallum 2005]. These three strategies are fast, and are commonly used as *uncertainly sampling* active learning strategies [Bernard, Zeppelzauer, Lehmann et al. 2018]. For the robustness of the experiment, in Chapter 5, all three techniques are included in the comparison with interactive visual techniques.

Only a few existing techniques work independently of the learning ML model, by choosing to focus on data characteristics. Some approaches explicitly allow users to select records in the kind of interactive visualisations typically used for data exploration or analysis [Bernard et al. 2014; Ritter et al. 2018]. For example, Heimerl et al. [2012] incorporates active learning for interactive visual labelling of text documents. Höferlin et al. [2012] introduced inter-active learning, which extends active learning to a visual analytics process for building ad-hoc training classifiers. The visual interactive-labelling (VIAL) process [Bernard, Zeppelzauer, Sedlmair et al. 2018] combines both model-based active learning and interactive visual interfaces to support the human-centered selection and labelling of records. Recent experiments have shown that individual strategies have different complementary strengths [Bernard, Hutter et al. 2018; Bernard, Zeppelzauer, Lehmann et al. 2018].

2.2.5 Evaluation of Visual Analytics Systems

There are many ways to evaluate interactive systems for visual analysis [K. Andrews 2006; K. Andrews 2008]. Lam et al. [2012] systematically reviewed over 800 visualisation publications and identified seven scenarios (motivations) for the evaluation of information visualisations: three for understanding data analysis processes and four for evaluating the visualisations themselves. Sedlmair, Meyer et al. [2012] identified nine stages of design when designing visualisations for domain experts. According to these scenarios and design stages, Wong et al. [2018] suggested appropriate evaluation methods for each stage.

It can, however, be challenging to evaluate such systems [Plaisant 2004; Carpendale 2008; Crisan and Elliott 2018]. Running controlled experiments on interactive visual systems can be particularly challenging. Datasets can vary wildly and tasks are often dependent on the kind of data being explored. Domain experts can be hard to find or unwilling to participate [Wong et al. 2018]. It is also hard to measure and compare the "insights" which such systems are designed to discover [North 2006].

The difficulty of running controlled experiments has lead to the increasing use of qualitative evaluation methods involving case studies and (longer term) observation of individual users. Shneiderman and Plaisant [2006] introduced the idea of the Multi-dimensional In-depth Long-term Case (MILC) study, a structured process to evaluate a VA system by observing a small number of domain experts using the system with their own datasets over a longer period of time. The MILC method has been shown to give a comprehensive understanding and high-quality results [Valiati et al. 2008; Perer and Shneiderman 2009].

2.2.6 Delineation of the Approach and Novelty in Chapter 4 and Chapter 5

The mVis tool which is introduced in Chapter 4 extends the approach of VIAL: analysts can use linked interactive visualisations to help mitigate the cold start problems associated with active learning. In addition, clustering and classification are provided to better guide the user in the labelling task. This novel technique lets the analyst intuitively build an ML model without the need for in-depth knowledge for machine learning algorithms. Moreover, Chapter 4 describes two stripped-down case studies of the mVis system with domain experts (in the spirit of MILC), which will be used to guide and inform future development and evaluation. In Chapter 5, it is shown how the proposed techniques surpass active learning algorithms, for labelling a multivariate dataset. As labelling is the first step to build an ML model for a dataset, any slight improvement in labelling process significantly reduces the cost of building an ML model and increases the accuracy of it.

2.3 Multimodal Interaction for Visual Analytics

The interactions in VA applications are often used as a direct medium to create queries and change the visualisation (e.g., for details on demand and changing views) using traditional WIMP interfaces. Moreover, they mostly focus on single modality (i.e. input device), to interact with the system. However, recent technology has brought novel sensing devices which allow for capturing user input and indirect user feedback beyond the typical desktop environment, e.g., by using eye-tracking. This allows for novel approaches for taking into account indirect user feedback (e.g., relevance feedback) and uses it to support and enhance the analysis process, both for making queries and as an indirect input for machine learning algorithms and VA techniques [Collins et al. 2018]. These input modalities are already proven to be beneficial in other domains such as Human-Computer Interaction and Computer Graphics [Turk 2014].

At a high level, information visualisation systems consist of two components: visual representation and interaction. Visual representation concerns the mapping from data to display [Yi et al. 2007]. The interaction starts with a user's intent to perform a task, followed by a user action. The system then reacts and feedback is given to the user [B. Lee et al. 2012]. It is essential to consider both visual representation and interaction when designing an application for information visualisation.

Research on combining multiple interaction modalities has a long history in the field of human-computer interaction. In one of the earliest works, Wang [1995] proposed VisualMan, a device and application-independent pipeline to integrate various modalities including gaze and voice into a user interface for 3D object selection and manipulation.

Some researchers have begun to look at how to integrate multiple interaction modalities into visual analytics interfaces. Srinivasan and Stasko [2018] presented Orko to explore the idea of using natural language interaction for network exploration. Shao, Silva et al. [2017] suggested using eye-tracking as input for exploration of patterns in scatterplot spaces. They used eye-tracking to detect which plots have been inspected by the user to suggest the most dissimilar plot by a guideline.

2.3.1 Visualisation on Large Displays

Researchers in various fields are increasingly confronted with the challenge of visualising and exploring high-dimensional datasets [Keim 2002; Shao, Mahajan et al. 2017]. Keim argues that although many traditional techniques exist to represent data, they are often not scalable to high-dimensional datasets without suitable analytical or interaction design [Keim 2002].

With the current size and resolution of typical computer displays, it is challenging to represent entire datasets on one screen using techniques like SPLOM or parallel coordinates. The user is often forced to resort to panning and zooming, leading to frustration and longer task completion times. Ruddle et al. [2015] conducted an experiment in which participants searched maps on three different displays for densely or sparsely distributed targets. They concluded that since the whole dataset fits on a larger display, sparse targets can be found faster.

Every view in a multiple linked views occupies space on display. If more space is available, additional views can be shown simultaneously. Allowing the user to access multiple windows increases performance and satisfaction [Czerwinski et al. 2003]. P. Isenberg, Dragicevic et al. [2013] present hybrid-image visualisation for data analysis, where two images are blended to achieve distance-dependent perception. This concept might be especially helpful for collaborative visual analysis tasks on vertically-mounted displays, where users observe data from various distances.

2.3.2 Visual Data Analysis and Multi-Touch Interaction

Previous researchers proposed various interaction techniques for large displays and multi-dimensional dataset interaction on multi-touch displays. Roberts [2005] proposed a classification of large display interaction having five dimensions: visualisation technology, display setup, interaction modality, application purpose, and location. Khan presented a survey of interaction techniques and devices for large, high-resolution displays [Khan 2011]. The survey categorises modalities of interaction into speech, tracking, gestures, mobile phones, haptic and other technologies such as gaze and facial expression.

Tsandilas et al. [2015] presented sketchsliders, a tool that provides a mobile sketching interface to create sliders which interact with multi-dimensional datasets on a wall display. Zhai et al. [2013] introduced gesture interaction for wall displays based on the distance of the user from the screen. The gestures can be performed in far or near mode. Heilig et al. [2010] developed multi-touch scatterplot visualisation on a tabletop display. Sadana and Stasko [2016] proposed advanced techniques for scatterplot data selection on smaller touch-based devices, such as tablets and smartphones.

MultiLens supports various gestures for fluid multi-touch exploration of graphs [Kister et al. 2016]. The Regression Lens [Shao, Mahajan et al. 2017] allows the user to interactively explore local areas of interest in scatterplots by showing the best fitting regression models inside the lens. The idea of visualising local regression models is also studied by Matković et al. [2017]. Rzeszotarski and Kittur [2014] introduced Kinetica, a tool for exploring multivariate data by physical interactions on multi-touch screens. Kister et al. [2016] presented BodyLenses, a promising set of magic lenses for wall displays,

which are mostly controlled by body interaction and therefore suitable for interacting with wall displays from a distance.

2.3.3 Collaborative Visualisation

Large displays are well-suited to collaboration [C. Andrews et al. 2011; P. Isenberg, T. Isenberg et al. 2013]. Jakobsen and Hornbæk [2014] conducted an exploratory study to understand group work with high-resolution multi-touch wall displays. The study suggests that using this kind of display helps users to work more efficiently as a group and fluidly change between parallel and joint work. A large display benefits group working on a shared task, since users can operate on one common physical medium and share information on it.

Morris et al. [2006] formalised the concept of cooperative gestures as a set of gestures performed by multiple users and interpreted as a single task by the system. Liu et al. [2017] developed CoReach, a set of gestures for collaboration between two users over large multi-touch displays. Comparing the use of a large vertically-mounted display against two ordinary desktop displays, Prouzeau et al. [2017] concluded that groups obtain better results and communicate better on large, vertically-mounted displays.

An experiment by Pedersen and Hornbæk [2012] showed that users prefer horizontal surfaces over vertically-mounted displays, but this result was limited to simple single-user tasks and not collaborative tasks with different dynamics. Vertically-mounted displays allow users to obtain an overview of their data by stepping back from the display and make it possible to interact from afar as well as up close. Badam et al. [2016] proposed a system for collaborative analysis on large displays by controlling individual lenses through explicit mid-air gestures.

2.3.4 Second Handheld Device

Using direct input on vertically mounted displays can cause the “gorilla arm” problem — a term used to explain the fatigue which sets in when users interact with their hands on a vertical screen for a prolonged period [Goodwins 2008].

Using a secondary handheld device is another option to interact with a large display, especially in situations where interacting with the display up-close is not adequate. Kister et al. [2017] designed GraSp – a set of spatially aware techniques for graph visualisation and interaction. Using these techniques, the analyst can explore graphs on wall displays by using a touchscreen on the secondary handheld device and body movement together with spatially aware mobile interactions. Later, Langner et al. [2018] presented a coordinated views application which can be controlled by both direct multi-touch and one or more secondary handheld devices. However, the secondary handheld devices are purely used for input, not for output.

2.3.5 Delineation of the Approach and Novelty in Chapter 6

In comparison to applications presented in Chapter 6, the aforementioned studies either focus on a different type of interaction and medium or are not designed for collaborative visual analytics tasks. Moreover, other studies are focused on less complicated visual analytics tasks, and do not tackle the problem of collaborative analysis of ML models, such as regression. This thesis presents two novel techniques for collaborative analysis of regression models on either large multi-touch display, or a combination of a handheld device and a large display. Moreover, Chapter 6 proves that gaze input can be used for visual exploration of multivariate datasets.

Chapter 3

Local Scatterplot Patterns

> *"There is in all things a pattern that is part of our universe. It has symmetry, elegance, and grace - these qualities you find always in that the true artist captures."*
>
> [Frank Herbert, Dune.]

The first step to incorporate VA into machine learning models is to use it for finding structures in the ML model. Incorporating VA for finding global structures are extensively researched, but searching for local patterns are often neglected. The novelty of this chapter is to use VA for finding local patterns in scatterplot spaces.

3.1 Introduction

The properties of a SPLOM for the purpose of data analysis have already been widely explored in the literature. While these techniques focus on exploring patterns within the data at a global level, little attention has yet been paid to the analysis of local structures and patterns. For the purposes of this work, a *pattern* is defined as a set of points in a scatterplot (i.e. from two of the n dimensions) contained within a specified bounding box. Points outside the bounding box are not part of the corresponding pattern. A *global pattern* comprises all the points in a particular scatterplot; in other words, the bounding box covers the entire scatterplot. A *local pattern* comprises a subset of points in a scatterplot. A *query pattern* is a pattern defined by the user, typically by interactively dragging a box.

One or more *descriptors* can be defined to characterise a pattern. A descriptor is a function taking a pattern (set of points) as input and producing a feature vector as output. *Shape-based descriptors* are based on the visual properties of the pattern, for example by subdividing the pattern into grid cells and calculating features such as the relative density of points in each cell. *Model-based descriptors* are derived mathematically, for example from a regression model, where individual features might be determined by evaluating the regression function at specific points.

The similarity between patterns can be defined in terms of similarity in the feature space of each corresponding descriptor, based on a distance metric such as the L1 metric or quadratic form distance [Beecks et al. 2010]. In practice, best results were often achieved using a similarity function defined as a weighted combination of distance metrics, subject to minimum thresholds for two measures of pattern matching quality. This is discussed in detail in Section 3.3.

With regard to local pattern analysis in scatterplots, Shao, Mahajan et al. [2017] previously proposed a scheme to explore and display regression models for interactively selected local regions of a scatterplot. However, that work only considered a single scatterplot (2 of the n dimensions) independently of any others. The implemented solution extends that approach to search for similar local patterns within

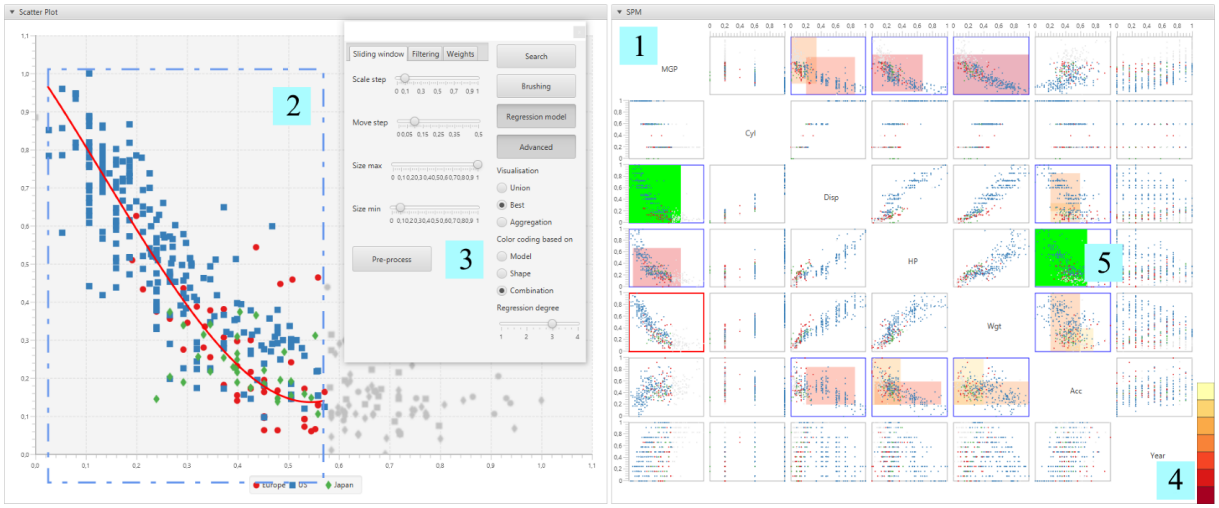


Figure 3.1: The SPLOM view ① can be examined for interesting patterns (rectangular regions). The colours of records (points) simply indicate the class they belong to. A query pattern can be specified interactively by dragging a bounding rectangle ② in a scatterplot of interest. The query pattern can further be adjusted using the floating toolbox ③. Matching patterns in the SPLOM are highlighted by coloured overlays according to their similarity to the query pattern. Dark red ones are the most similar and yellow ones are least similar ④. The patterns with the green overlay are patterns marked by the user as relevant ⑤.

all other scatterplots in the SPLOM. Matching patterns are ranked by similarity and presented to the user for further exploration. Visual highlighting is used to indicate matching local patterns within other scatterplots. The user can refine the search by selecting the patterns most relevant to their interests. A relevance feedback mechanism is then used to identify and recommend additional potentially relevant results. Figure 3.1 shows an overview of the approach.

For example, consider a dataset where the dimensions are attributes of a country (such as population, GDP, etc.) and the records are data from various countries and years (such as Japan 2010). Choosing an interesting local pattern and being able to explore similar local patterns within other scatterplots opens up a powerful new way to discover relationships between subsets of records across the entire multidimensional space. Appropriate interaction mechanisms allow the user to inspect the ranked set of matching local patterns and refine their query to explore further. While this work focuses on the search for local patterns in other dimensions, the approach can be easily extended to finding patterns in other scatterplots of the same dimensions. For example, consider a dataset containing information about customers in various quarters of the year. Each quarter can be shown as a scatterplot, allowing the analyst to find similar patterns in other quarters. Another use case is searching for similar patterns in one scatterplot and on different clusters. For example, in the countries dataset, one could be interested in countries following the same pattern in the GDP-Population scatterplot.

The main goal was to develop an approach to find local patterns in a scatterplot space, without prior knowledge about the dataset. The design of a search algorithm for defined local patterns differs fundamentally from the design of a system dealing with general use cases. If the patterns of a dataset are known, it is possible to add carefully tailor-made descriptors. If no prior knowledge about the dataset exists, a general algorithm and descriptors to search for patterns are needed. For this reason, an approach which takes into account shape and model-based descriptors, various parameters, and a relevance feedback module was introduced. The approach can be further customised for specific datasets. The contributions of this chapter are:

1. A set of interactive strategies to select local patterns of interest in one or more scatterplots.

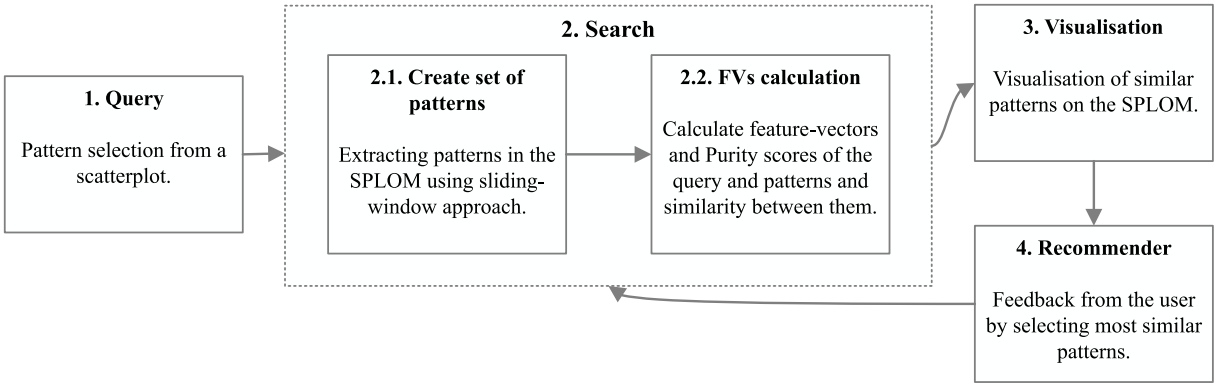


Figure 3.2: Finding similar local patterns in a SPLOM. First, the user selects a rectangular portion of a scatterplot to create the initial query ①. Next, all identifiable patterns are extracted from the SPLOM using a sliding-window approach. Their feature vectors and purity scores are calculated, leading to a similarity score between each pattern and the query ②. The most similar patterns are visualised in the SPLOM ③. The user can then indicate which of the matching patterns are most relevant to their needs, thus refining the query ④.

2. An approach facilitating local pattern exploration by suggesting similar local patterns in other scatterplots across the entire SPLOM, based on relevance feedback.

3. Shape-based and model-based descriptors to characterise local patterns.

4. A similarity metric to determine the best matching local patterns in the rest of the multidimensional space.

3.2 Research Questions and Hypothesis

After discovering an interesting local pattern in one scatterplot, an analyst sometimes wants to search for similar local patterns in the rest of the SPLOM to investigate otherwise hidden relationships such as correlations between dimensions in a subset of the dataset.

This chapter addresses RQ1: *How to use VA for finding structures in the ML model?* More specifically, how to use VA to effectively search within local structures in a scatterplot space. Therefore the following research question is asked.

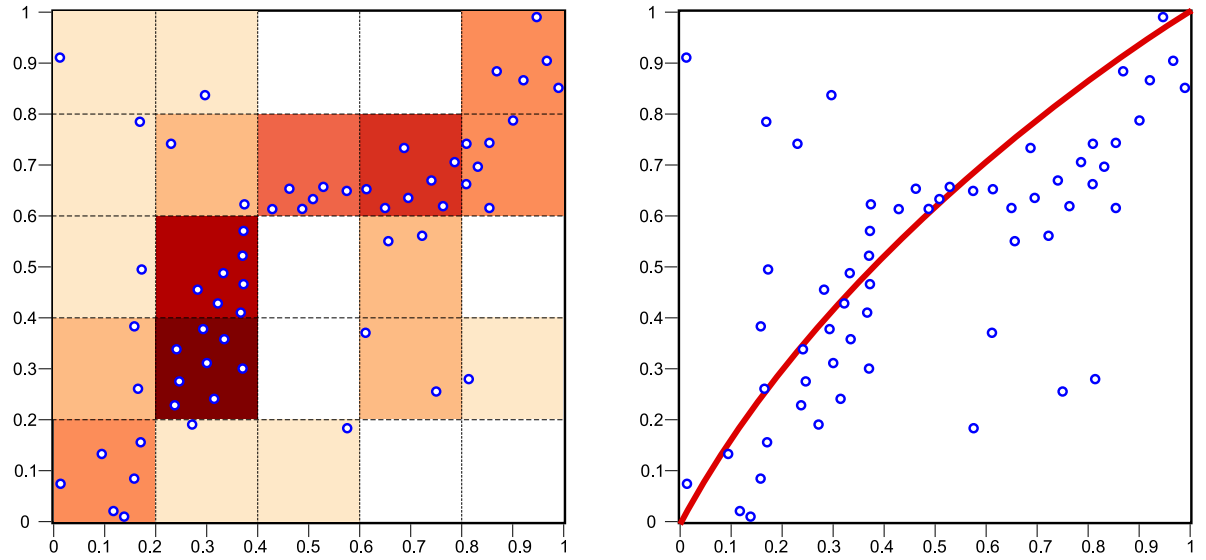
Research Question 3.1 (RQ3.1): *How to use VA for finding local patterns in a scatterplot space of a multivariate dataset?*

This chapter presents a novel visual analytics approach to select, search for, visualise and refine the search for local patterns. Figure 3.2 illustrates the search pipeline. Therefore, a hypothesis correspondent to RQ3.1 is formed.

Hypothesis 1 (H1): *Exploratory visual analytics, together with similarity search, is well suited for finding local patterns in scatterplot spaces.*

3.3 Exploring Local Patterns in Scatterplots

To begin a search for local patterns, the analyst draws a bounding box around a set of points in a scatterplot of interest, which specifies the initial query (Step 1 in Figure 3.2). The system then extracts a set of patterns from the entire space of scatterplots in the SPLOM. These patterns are built by successively



(a) A shape-based descriptor defined by point density in a 5×5 grid, where the cell colour indicates relative density.

(b) A model-based descriptor defined by a quadratic regression model.

Figure 3.3: Shape-based and model-based descriptors.

translating and scaling a bounding box over each scatterplot. The bounding box moves over a scatterplot by a discrete *translation step size* and scales from the smallest size until it fits the whole scatterplot by a discrete *scaling step size*. All the patterns generated are added to the resulting *set of patterns*, which holds all extracted patterns (Step 2.1).

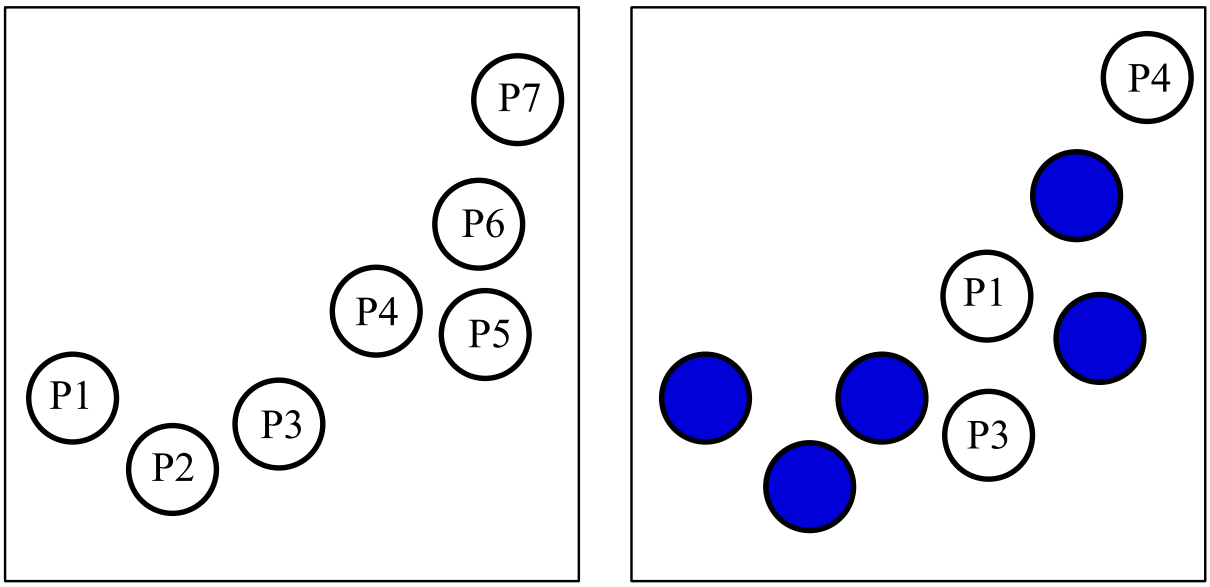
Feature vectors are extracted to describe each pattern using both model-based and shape-based descriptors. Shape-based descriptors represent the perceptual similarity of patterns. Since they characterise the data according to appearance, they have some limitations [Pandey et al. 2016]. Model-based descriptors (currently from regression models) are used to capture the relationship between points in a pattern (Step 2.2). Since it can also be important that both the query and a pattern from the set of patterns contain a larger set of identical records, an overlap of records in the query and the target pattern is computed, and is used to filter the results. To this end, purity scores (Section 3.3.2) are introduced. Purity scores indicate how many records from the query, exist in a pattern.

A ranking of patterns is determined by comparing the query and patterns using the descriptors, and the best matching patterns are then visually highlighted in the SPLOM (Step 3). The user can now select the patterns most relevant to their needs and the relevance feedback module refines the search parameters based on user's feedback and searches for new patterns (Step 4).

3.3.1 Model-Based and Shape-Based Descriptors

Shape-based descriptors use shape information to characterise a pattern. The pattern is partitioned into a grid of cells. Then, a 2D histogram is calculated, in which each feature represents the density of points in the corresponding cell. The density is calculated by $N_{subset} \,/\, N_{total}$, where N_{subset} is the number of points in the cell and N_{total} is the total number of points (records) in the pattern. Figure 3.3a shows a simple 5×5 grid of a pattern. This descriptor is scale-invariant.

Model-based descriptors characterise relationships between points in a pattern. For example, a local regression model estimates the relationship among variables in the local area of a scatterplot. Based on



(a) User-defined query pattern containing seven points.

(b) A matching pattern containing eight points, of which three are contained in the query.

Figure 3.4: A user-defined query pattern and one matching pattern

the notion of the regression model, a regression-based descriptor is used. Since ranges of the regression models are not the same, it is not possible to compare two feature vectors created by two different regression models unless values of points are normalised. For this reason and also to keep the descriptor scale-invariant, the points are normalised. Then, a linear, cubic, quadratic or 4th-degree regression model is calculated. Assuming the regression model is $f : X \rightarrow Y$ and $X \in \{0.0, 0.1, \ldots, 0.9, 1.0\}$, the feature vector is built by iterating over the domain X. Therefore, there are 11 features in the feature vector. Figure 3.3b shows an example of a regression model for a pattern. The function to calculate the distance between two feature vectors is:

$$\frac{\sum_{i=0}^{n}(|y_1^i - y_2^i|)}{n} \tag{3.1}$$

3.3.2 Purity Scores for Pattern Comparison

Since the set of records in the query and a pattern are not always equal, it is useful to consider the similarity of records. Inspired by pattern recognition algorithms, purity scores $P_{precision}$ and P_{recall} are introduced. The $P_{precision}$ is calculated by dividing the total number of records shared between both patterns, N_{shared}, by the total number of records in the target pattern, $N_{pattern}$. This score shows how many matched records exist in the target pattern. In contrast, P_{recall} is defined as N_{shared} divided by N_{query}, which is a total number of points in the query. This score indicates what percentage of records is repeated from the query in the target pattern:

$$P_{precision} = N_{shared} \; / \; N_{pattern} \tag{3.2}$$

$$P_{recall} = N_{shared} \; / \; N_{query} \tag{3.3}$$

Both scores are between zero and one. It is possible that the analyst prefers to filter out patterns with low purity scores. Figure 3.4 gives an example of purity scores. Figure 3.4a shows a query with seven records. Figure 3.4b shows a target pattern containing three records from the query. In this case, the purity scores are $P_{precision} = 3 \; / \; 7$ and $P_{recall} = 3 \; / \; 8$.

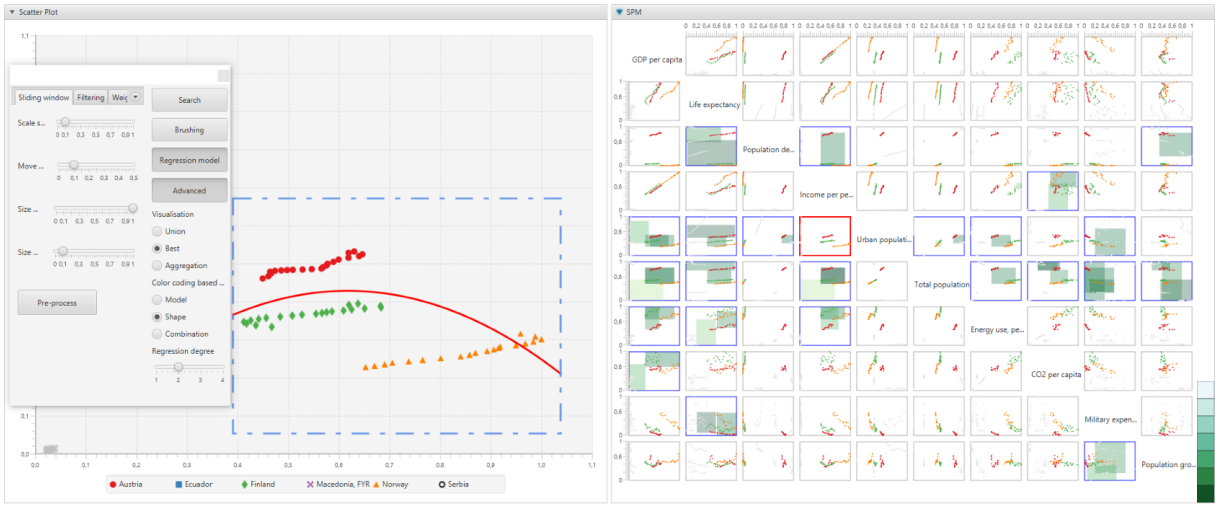


Figure 3.5: A snapshot of the application showing the Countries dataset from the World Bank [TWB 2018]. The scatterplot outlined in red in the SPLOM is selected and is shown enlarged in the scatterplot view to the left. The user has already specified a query pattern and chosen to visualise the patterns in aggregation mode using shape-based descriptor colour-coding. In aggregation mode, all patterns similar to the query are highlighted in the SPLOM. The greenish colour-coding indicates the strength of the similarity according to shape-based descriptors.

3.3.3 Ranking Algorithm

A ranking algorithm is used to obtain a ranked list of patterns similar to a query pattern. The ranking algorithm incorporates the distance between descriptors (Section 3.3.1) of the query and candidate patterns, as well as purity scores between them (Section 3.3.2).

In the first step, L_1-distances between descriptors of the query and all candidate patterns are calculated. For each descriptor type, separate distance measures of d_s (shape-based descriptor) and d_m (model-based descriptor) are obtained. Both of these distance sets are then min-max normalised to $[0, 1]$. For the ranking, both distances are combined with a parametrised weight and scaling. Subsequently, purity scores are computed among query and candidate patterns to filter the ranking.

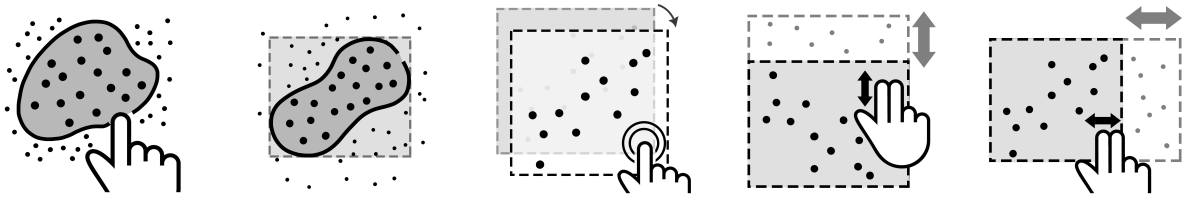
The effective ranking computed by the algorithm can be formalised as a descending ordered list with respect to the similarity score $s : (q, p)$ (Formula 3.4) between the query pattern q and each candidate pattern p:

$$s(q,p) = \begin{cases} 0 & \text{if } P_{precision} < P_{pmin} \text{ or } P_{recall} < P_{rmin} \\ w_m(1-d_m) + s(1-w_m)(1-d_s), & \text{otherwise} \end{cases} \tag{3.4}$$

where w_m and $1-w_m$ are weights assigned to the descriptors, P_{pmin} and P_{rmin} are minimum thresholds for purity scores, and s is an additional distance scaling coefficient. Scaling distances after they have been normalised might further improve ranking quality, since the distribution within the distance spaces of d_m and d_s is likely to differ significantly. This cannot be compensated for by min-max normalising both distributions. In practice, the greatest interest is directed towards patterns having the smallest distances to a query and not towards those patterns yielding the largest distances.

The ranking algorithm can be customised by adjusting several parameters in a graphical dialogue to better reflect the user's notion of similarity for the respective dataset (or domain) at hand:

$\boldsymbol{w_m}$ within $[0, 1]$ balances the weight between distances d_s and d_m obtained for each descriptor type.



(a) Selection **(b)** Lens **(c)** Moving **(d)** Scaling **(e)** Scaling

Figure 3.6: Possible user interactions with the scatterplot. (a) The user first draws an arbitrary shape around the points. (b) The bounding box appears around the records. (c) The user can translate the box. (d) By placing two fingers on the top side of the box, the user scales the box vertically. (e) By placing two fingers on the right side of the box, the user scales the box horizontally.

s is an arbitrary real-valued number that can be used to fine-tune the weighting. In particular, it might compensate distortion for the first results in case of very differently shaped distributions within the distance spaces obtained by d_s and d_m.

$\boldsymbol{P_{pmin}}$**,** $\boldsymbol{P_{rmin}}$ are thresholds in $[0, 1]$ that steer the filtering of the ranking by purity scores, where 0 disables the filtering. If both are set to 1, only candidate patterns sharing all points with the query will yield non-zero similarity.

shape-based descriptor resolution is an additional parameter within $\{2, 3, 4, 5\}$ which controls the spatial resolution (number of grid cells) of the shape descriptor.

model-based descriptor degree is an additional parameter which controls the degree of the polynomial regression used for the model-based descriptor. In the current system, it is within $\{1, 2, 3, 4\}$.

3.3.4 Relevance Feedback Algorithm

By examining the ranking algorithm above, the set of parameters involved in computing a similarity ranking of patterns for a respective query can be identified. For example, in Formula 3.4, the values for w_m, P_{pmin} and P_{rmin} can be any number in [0,1] and the coefficient s can be an arbitrary number. Moreover, four shape-based and four model-based descriptors are used, which affect d_m and d_s.

As mentioned in the previous section, the system provides a graphical dialogue, through which the user may tweak individual parameters of the ranking algorithm. However, for many users without a background in information retrieval, tweaking these parameters is difficult. Even with more in-depth understanding of the ranking algorithm, tweaking the parameters in a meaningful way is highly specific to the characteristics of the currently used dataset.

This issue is addressed by applying a relevance feedback module, which derives parameter values from user-provided examples of similar patterns. After an initial search, the module enables the user to select multiple patterns from the result set, which best match his or her notion of similarity. There is no need for manual parameter tuning, since the user provides feedback by indicating which patterns are most relevant to their current needs. In essence, the user tweaks the parameter set indirectly through relevance feedback. This method is widely used in information retrieval systems.

The relevance feedback module then evaluates the rankings of the selected result patterns with a large number of parameter configurations and selects the configuration C_k which minimises the aggregated rankings of the selected patterns. The operation is shown in Formula 3.5, where $u_1, u_2, \ldots, u_j$ are the patterns selected by the user and $ranking_{C_k}(q, u_i)$ denotes the ranking position obtained for pattern u_i and

query q when a certain parameter configuration C_k is used (see Section 3.3.3):

$$\underset{C_k}{\operatorname{argmin}}\left(\sum_{i=1}^{n} ranking_{C_k}(q, u_i)\right) \tag{3.5}$$

To keep the delay in the graphical user interface to a minimum and also to avoid numerical issues in certain cases, the problem is not addressed continuously. Instead, all aggregated ranking scores are computed over a discrete set of possible parameter configurations for the user selected set of similar patterns. Since both the descriptors and the respective distances can be precomputed, a ranking can be computed with almost negligible computational cost (see Section 3.3.5 for details). It is hence possible to evaluate several thousand parameter configurations without noticeable delay in the graphical user interface.

In the current system, all configurations C_k are obtained as the 6-fold Cartesian product of the discrete value ranges for the individual parameters of the ranking algorithm. In addition to the value constraints for the parameters that are already mentioned in Section 3.3.3, the weights w_m used for evaluation are within $\{0.0, 0.2, 0.4, 0.5, 0.6, 0.8, 1.0\}$ and the scaling s is within $\{1.0, 1.5, 2.3, 3.4\}$. The numbers are chosen heuristically. In the current implementation, only a single value is used as range for P_{pmin} and P_{rmin}, but both are computed based on the user selected set of patterns according to Formula 3.6 and 3.7. In these equations, 0.2 is a number derived heuristically from various search results:

$$P_{pmin} = \min_{u_i}\Big(P_{precision}(q, u_i)\Big) - 0.2 \tag{3.6}$$

$$P_{rmin} = \min_{u_i}\Big(P_{recall}(q, u_i)\Big) - 0.2 \tag{3.7}$$

3.3.5 Complexity of the Algorithm

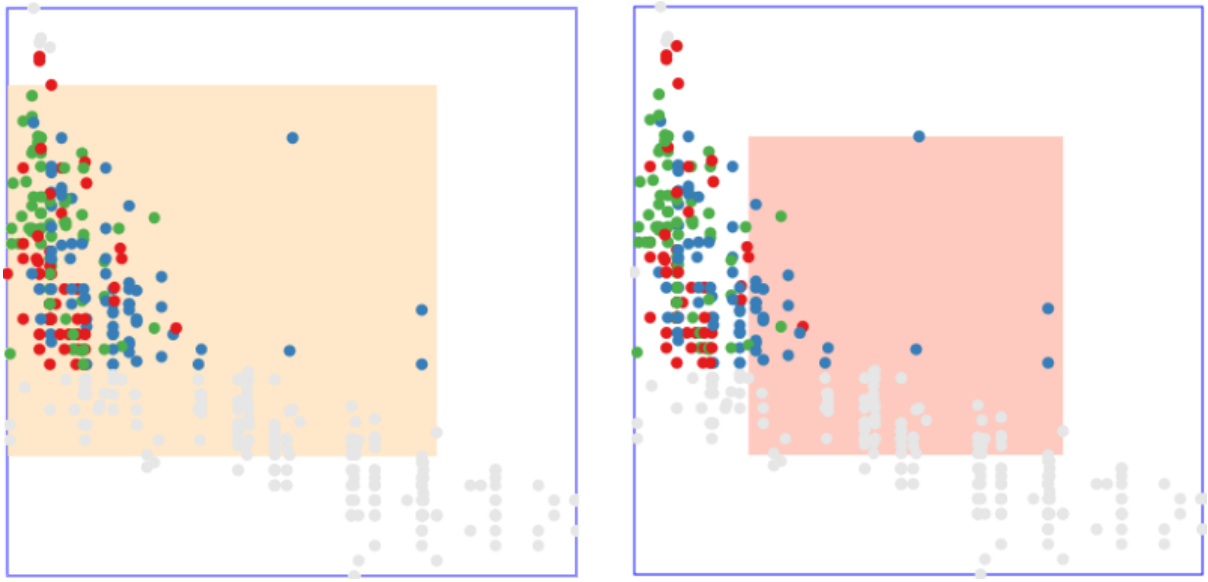
The run-time of the algorithm has two aspects. The first is the pre-processing required for a particular dataset. The second is the time required to search for patterns during user interaction. For the first aspect, three variables play a role:

Step size for sliding window: Both the translation step size and the scaling step size directly affect the number of patterns extracted per scatterplot. For example, if the translation step size is 0.1 and the scaling step size is 0.2, at most 385 patterns are extracted. In practice, some windows do not include any data points and therefore the number of extracted patterns is reduced.

Number of dimensions: If a data set has n dimensions, the maximum number of extracted patterns is $n \times (n-1) \times 385$. For the example step sizes above, a dataset with 10 dimensions would see at most 34650 patterns extracted.

Number of descriptors: For each pattern, a number of feature vectors are extracted. This number directly affects the computation time. In this work, four feature vectors are used for shape-based descriptors and another four for model-based descriptors. If calculating a feature vector for a pattern costs, say, time t_d, the final maximum time for pre-processing the example dataset above would be $34650 \times t_d$.

For the second, run-time aspect of the algorithm during user interaction, the number of parameters and patterns extracted in the pre-processing stage play an important role. For the configuration described in this chapter, $7 \times 4 \times 7 \times 4 \times 4$ searches are conducted per pattern. If t_s is the time for one search, then a maximum time of $34650 \times 3136 \times t_s$ is required for one query.



(a) Search with purity scores. **(b)** Search without purity scores.

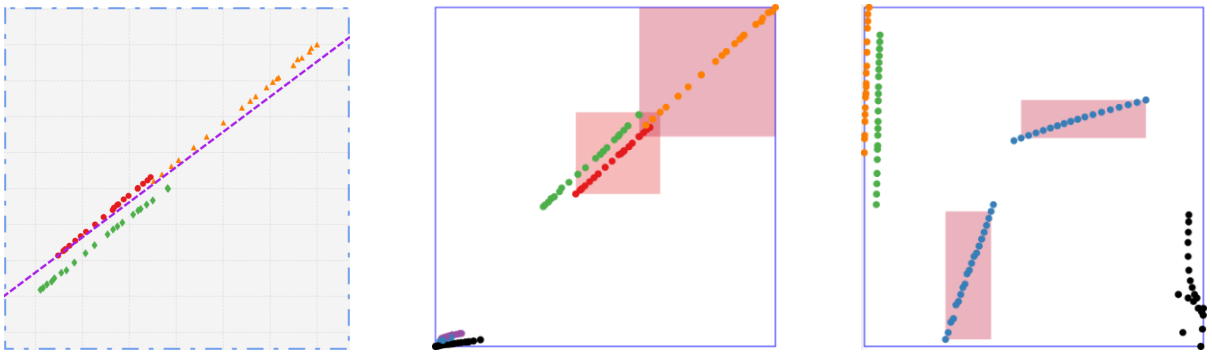
Figure 3.7: The coloured records (red, green, and blue) are chosen by the user in the query. In (a), the user chooses to filter only patterns having a high similarity of records with the query, while in (b) there is no restriction.

3.4 System Overview

To answer RQ3.1, a prototype application has been implemented on a vertically-mounted Eyevis 84-inch multi-touch display with a resolution of 3840 × 2160 pixels and a frame rate of 60 Hz [eyevis 2018]. This setup is being extended for multimodal, multi-user scenarios. The prototype application is written in Java, using JavaFX for the user interface and the TUIO [Kaltenbrunner et al. 2005] and TUIOFX library [Fetter and Bimamisa 2015] for multi-touch interaction. The application consists of two linked views, a scatterplot, and a SPLOM. The user can open a scatterplot from the SPLOM in a new window or in the existing scatterplot view. The records in scatterplots and the SPLOM are coloured based on their (predetermined) class labels. By using a large, high-resolution display, visualising a multi-dimensional dataset on a SPLOM is supported.

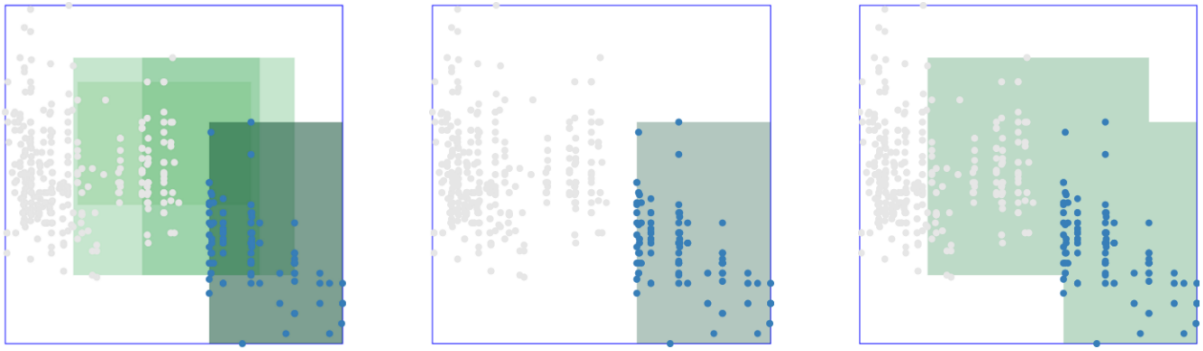
3.4.1 Constructing a Query

To construct a query, the analyst first selects a scatterplot from the SPLOM. The selected scatterplot is shown on the left panel. As shown in Figure 3.6, the analyst draws an arbitrary closed shape in the scatterplot to select a set of records (points). While previous studies suggest creating the search query based on a sketch [Shao et al. 2014], this proposed free-form selection technique enables the user to search for local patterns. The query is built by fitting the minimum sized rectangle around the selected records. Inspired by the work of Shao, Mahajan et al. [2017], a regression model is visualised within the rectangle as well. In this work, the regression model is used to help the analyst obtain an abstraction of the pattern for a better understanding of the final query. The abstraction of information is a significant step in information visualisation to reduce cognitive efforts to interpret the data [Gleicher et al. 2011]. The analyst can scale the rectangle to include more records. Moreover, for more fluid interaction with large multi-touch screens, a floating toolbox is provided as shown in Figure 3.5. The analyst can manipulate the rectangle with one hand and the floating toolbox with the other hand simultaneously. Once selection of query points is finished, the analyst taps the search button to initiate a search.



(a) The query. **(b)** First scatterplot. **(c)** Second scatterplot.

Figure 3.8: (a) The user selects a query. Since both shape and model-based feature vectors are scale-invariant, all four patterns in scatterplots (b) and (c) are marked as similar.



(a) Aggregation **(b)** Best-matches **(c)** Union

Figure 3.9: Three different techniques are used to highlight matching patterns in a scatterplot. In (a), all matching patterns are shown in aggregation. In (b), the best matching patterns are highlighted. In (c), the union of all matching patterns is highlighted.

3.4.2 Search

After query selection, the system searches for similar patterns in the SPLOM. Patterns in the SPLOM are extracted and all feature vectors are pre-calculated in multiple threads as the dataset is loaded into the application. On a standard PC with an Intel Core i7 CPU and 18 GB of RAM, a dataset containing ten dimensions, 240 records, producing a total of 5941 patterns, requires 26 seconds for pre-calculation. The calculated values are stored in a cache. Each pattern contains four shape-based and four model-based feature vectors.

3.4.2.1 Set of Patterns

To achieve better performance, a set of patterns is created from the SPLOM just once after the dataset is loaded. A set of patterns per scatterplot is generated by the sliding-window approach. A pattern with less than 5 points is ignored. For a dataset containing 240 records and ten dimensions, the system extracted 5941 patterns. The translation and scaling steps of the window are adjustable by the analyst. As shown in Figure 3.5, the analyst can change the step sizes by clicking on the Advanced button in the toolbox.

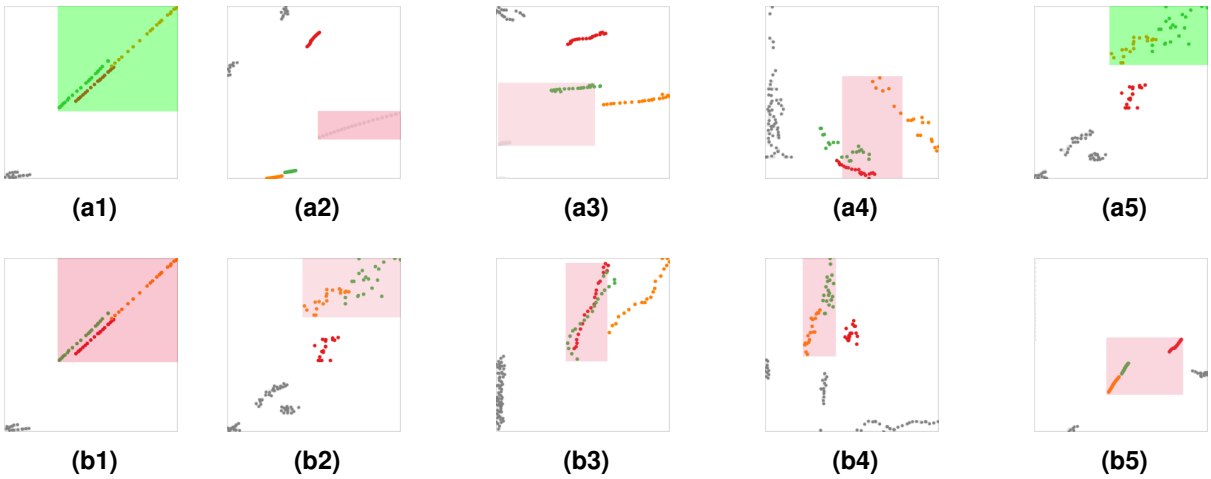


Figure 3.10: The results of the query shown in Figure 3.8a. The first row shows the best matches in the initial search results, where the weight of both shape-based and model-based descriptors are the same. The second row shows the five best matches after the user chose the first and the last matches in the initial search result and the relevance feedback module has adjusted the search parameters accordingly.

3.4.2.2 Purity Scores and Feature Vectors

The Advanced options in the floating toolbox allow the analyst to manipulate the parameters of the ranking algorithm. The options are located in the Filtering and Weights tabs. For example, in Figure 3.7(a), the analyst chose 50% for Min $P_{precision}$ value in the Filtering tab, while in Figure 3.7(b) the value is set to 100%. Instead of configuring the parameters manually, the analyst can rely on the relevance feedback module.

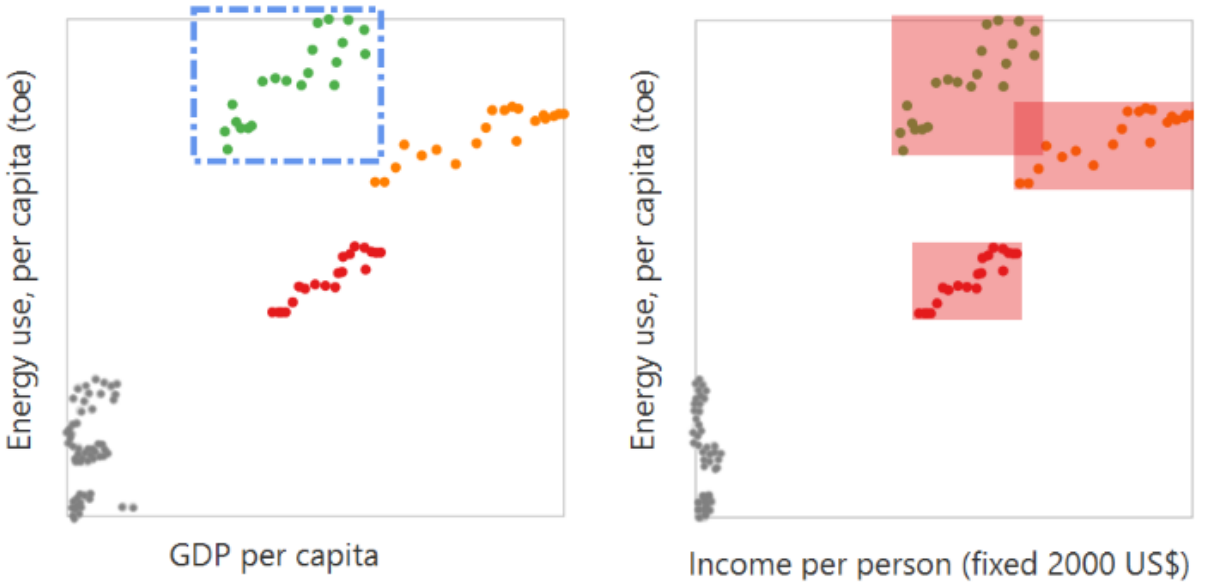
3.4.3 Similarity Visualisation

After the similarity search algorithm has determined the distance between items in the set of patterns and the query, the application visualises the patterns. Three techniques are used to visualise them: visualising a differing number of patterns based on the user's need, interactive brushing, and colour-coding based on distance.

Firstly, the user can choose between three options to manage the number of visually highlighted patterns: aggregation, best-matches, and union. As shown in Figure 3.9a, by choosing aggregation, all similar patterns according to the similarity search algorithm are visualised. This method may show overlapping patterns and creates rectangles that do not exist. To avoid this, the user can select the best-matches option, shown in Figure 3.9b. In this method, if two rectangles overlap by more than 70%, the more similar pattern will remain. The union in Figure 3.9c combines all patterns into one shape and the colour of the shape is the average of patterns combined.

Secondly, by brushing the selected points in scatterplots, the similarity scores are visualised. This method makes the user aware of the similarity between selected records and patterns.

Pandey et al. [2016] showed that judging similarity between plots purely according to their appearance may be misleading. Therefore, thirdly, to avoid relevance feedback favouring shape-based descriptors, a colour-coding function to show similarity according to different descriptors was implemented. Three multi-hue colour palettes are used to indicate the distance between the query feature vectors and pattern feature vectors based on shape-based descriptors (green shades), model-based descriptors (purple shades), and their combination (red shades). Figure 3.1 and Figure 3.5 show the colour palettes.



(a) The initial query (Finland) in the scatterplot of Energy Use against GDP per Capita.

(b) Similar patterns are found in the scatterplot of Energy Use against Income per Person.

Figure 3.11: Three local patterns are found, indicating similar behaviour of three countries in another scatterplot.

3.4.4 Relevance Feedback Module

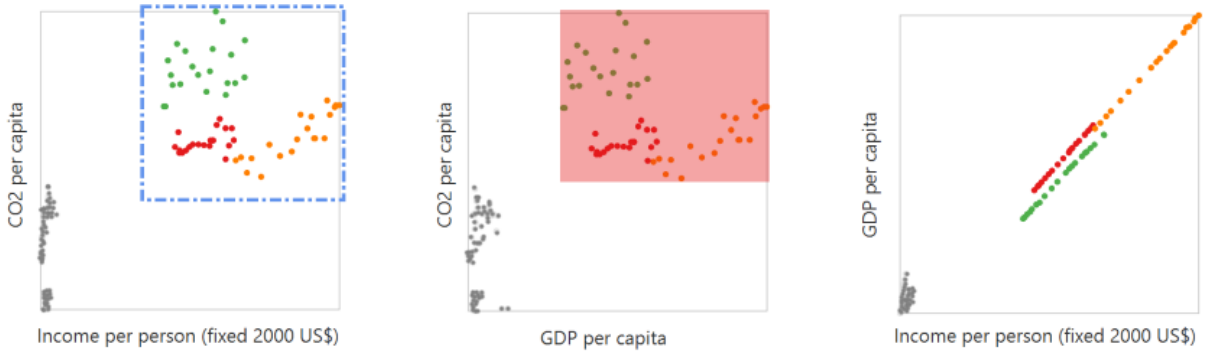
As an example of relevance feedback, the top row of Figure 3.10 shows the top five query search results from the query in Figure 3.8a. The second row shows the refinement after relevance feedback. Records in the query have a positive linear relation. In the first row, the system declares five patterns as most similar to the query, whereby $w_m = 0.5$, $P_{pmin} = 0.00$ and $P_{rmin} = 0.00$ were used. The grid size of the shape-descriptor is 2×2 and the regression model in the model-descriptor is quadratic. The user indicates that the first and fifth patterns (Figure 3.10a1 and Figure 3.10a5) are relevant and the patterns highlighted in green.

The relevance feedback module determines a new set of parameters in which $w_m = 0.2$, $P_{pmin} = 0.80$ and $P_{rmin} = 0.57$ with a grid size of 3×3 for shape-based descriptors and linear regression model for model-based descriptors. The pattern in Figure 3.10a2 is ignored since the similarity score thresholds are not satisfied. The patterns in Figure 3.10a3 and Figure 3.10a4 are similar to the query by the model-based descriptor, but since the weight for it is low, they are taken out of the ranking. Also, the $Purity_{recall}$ and $Purity_{precision}$ are below the thresholds. As shown in the second row of Figure 3.10a3, the new patterns are more visually similar to the query.

This example shows the usefulness of the relevance feedback module. Since the search algorithm is scale-invariant, some found patterns have a significantly different slope to the query. The system does not know the meaning behind units within scatterplots, therefore the scaling and angle of slope may not be meaningful.

3.5 Use Cases

Any multivariate dataset can be loaded into the application. To test Hypothesis 1, three query scenarios are discussed using a subset of the Countries dataset from the World Bank [TWB 2018]. The datasets contains ten dimensions and 126 records. The dimensions are attributes of countries such as GDP per



(a) The initial query.

(b) A similar pattern in another scatterplot.

(c) A local positive correlation.

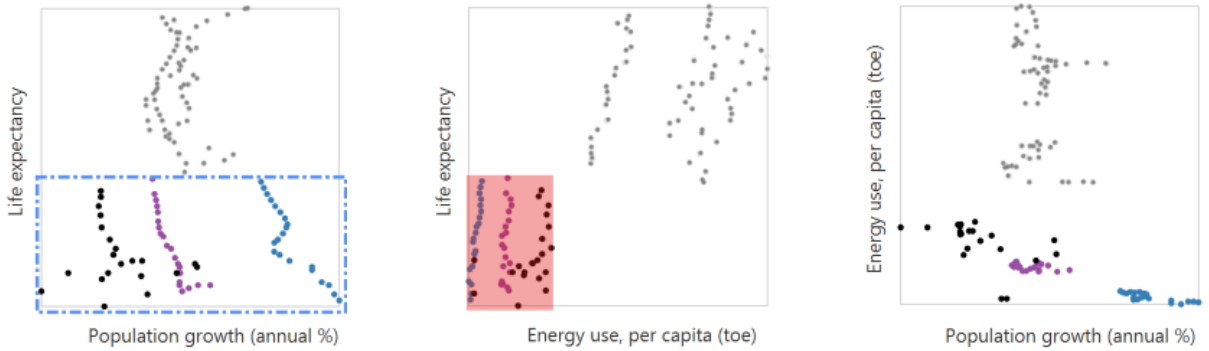
Figure 3.12: Since the result pattern looks very similar to the initial query, a possible relationship might be expected between the two horizontal scatterplot dimensions.

Capita, Life Expectancy, and Population Growth. The records contain information about each country for each year between 1995 and 2015. Here, the data for six countries was used: Austria, Ecuador, Finland, Macedonia, Norway and Serbia. Figure 3.5 shows the dataset as visualised in the system. All three query scenarios were refined using the relevance feedback module. For simplicity, only the initial query and final results are presented.

The first query scenario is shown in Figure 3.11. The user selected all data points corresponding to Finland (green) in the scatterplot of Energy Use plotted against GDP per Capita. After refining the search with relevance feedback, three similar patterns belonging to Austria, Finland, and Norway were found in the scatterplot of Energy Use against Income per Person. This result is perhaps to be expected, since all three nations are developed countries in Europe and GDP and Income per Person are generally related. The parameters of the search are $P_{pmin} = 0.00$, $P_{rmin} = 0.00$, $s = 1.0$, $w_m = 0.5$, 4×4 resolution of the shape-based descriptor, and a linear model-based descriptor.

After investigation of the first query, the user decides to investigate the scatterplot of CO2 per Capita against Income per Person. This time, the user selects all three aforementioned countries to form the initial query, as shown in Figure 3.12. As may have been suspected, the same pattern is found in the scatterplot of CO2 per Capita against GDP per Capita. Again, the user suspects a possible relationship between Income per Person and GPD per Capita. The assumption is valid since there is a local positive correlation for these three countries, as shown in Figure 3.12c. After checking with other similar queries, the user concludes that the Income per Person dimension is redundant and that keeping GDP per Capita is sufficient for their purposes. The parameters of the search are $P_{pmin} = 0.80$, $P_{rmin} = 0.80$, $s = 1.0$, $w_m = 0.2$, 3×3 resolution of the shape-based descriptor, and a quadratic model-based descriptor.

The user continues to explore the dataset by examining the scatterplot of Life Expectancy against Population Growth, as shown in Figure 3.13a. The user selects the data points comprising Macedonia, Serbia, and Ecuador at the bottom of the scatterplot. The most similar pattern is located in the scatterplot of Life Expectancy against Energy Use. This pattern looks like a flipped version of the query. The user decides to inspect the scatterplot of Energy Use against Population Growth to look for any local relationship between records in the query. As shown in Figure 3.13c, a local negative correlation exists between the points of these three countries, but no correlation is apparent when all of the points in the scatterplot are considered. The parameters of the search are $P_{pmin} = 0.59$, $P_{rmin} = 0.65$, $s = 1.0$, $w_m = 0$, 5×5 as the resolution of the shape-based descriptor, and a quadratic model-based descriptor. Since $w_m = 0$ in this search, the model-based descriptor is ignored.



(a) The initial query.

(b) A similar pattern in another scatterplot.

(c) A local negative correlation.

Figure 3.13: There is some similarity between the initial query on the left and the matching local pattern on the right. There is a local negative correlation between the two horizontal scatterplot dimensions.

3.6 Discussion

The presented approach allows users to search for similar local patterns in a set of scatterplots, helping users explore multidimensional datasets by comparing patterns. More specifically, the approach focuses on finding related patterns with regard to shape-based and model-based similarity across different regions, dimensions, and record subsets of a larger SPLOM space. Therefore, this chapter answered RQ1, and more specifically RQ3.1.

Users can initiate a search by interactively selecting a region of a scatterplot as an initial query. An obvious extension would be to include a sketch-based interface where query patterns can be sketched in free form. Currently, the user must manually inspect the SPLOM to find a suitable query pattern in a scatterplot. A more scalable approach for a larger SPLOM would be to include a clustering step to identify representative local patterns (e.g., using density-based clustering). Then, an overview of cluster prototypes could be offered to the user to choose a query pattern, optionally editing this using sketching or by blending with other prototype patterns.

Through experimentation, it was found that similarity search of scatterplot patterns depends on the chosen descriptors and dataset. When inappropriate descriptors are used, the search results may be perceived as dissimilar by the user, although they are similar according to the definition of the descriptors. A relevance feedback approach allows users to tune search parameters implicitly, steering the system towards a notion of relevance fitting their current needs. In informal experiments, it was observed that this approach can return more relevant patterns. More formal evaluation regarding information retrieval measures would be interesting. The usability of the relevance feedback interface needs to undergo some more formative evaluation. In addition, a comparative study of the system with and without relevance feedback would be desirable.

Finally, the use cases provided in this chapter tested and proved Hypothesis 1. It is shown that exploratory VA is well suitable to find local patterns in scatterplot spaces, such as local positive, or local negative correlations.

Chapter 4

Interactive Visual Labelling

"Once you label me you negate me."

[Søren Kierkegaard, Danish philosopher, 1813 - 1855]

After incorporating VA for finding local patterns in scatterplot spaces, the next question is how to use it for building ML models. This chapter introduces a novel interactive visual technique, to enable the analyst to build an ML model of a multivariate dataset for supervised ML, using linked visualisations, clustering, and active learning.

4.1 Introduction

A multivariate dataset is a dataset with more than one dimension. Partitioning a multivariate dataset into labelled classes (partitions) is one of the most prominent supervised machine learning (ML) tasks. Every record in a partitioned dataset must belong to exactly one of the partitions: records cannot belong to multiple partitions, nor can they be left belonging to no partition.

Once a classifier has learned the characteristics of a given multivariate dataset in the training process, the ML model can thereafter be used to automatically partition other, similar datasets. The state of the art in ML demonstrates the effectiveness of today's classifiers in many domains, from the detection of attacks in computer networks [Lin et al. 2017] to facial image data analysis [Choo et al. 2010].

Two prerequisites for effective ML techniques are the availability of (1) sufficiently large training datasets and (2) labels provided with those datasets. Without labels, a supervised ML model cannot be trained. Without sufficient numbers of labelled records for training, the supervised ML model will not perform effectively.

However, the unavailability of labels for many real-world datasets is often the bottleneck in supervised ML applications. Today's scientists are often overwhelmed by thousands or even millions of unlabelled records in datasets, all of which are thus unavailable for supervised ML. Given a means to more effectively support analysts in the labelling process, a plethora of unsolved real-world data-centered challenges could be addressed with ML techniques.

The particular challenge addressed by the approach can be exemplified by a domain expert wanting to use a previously unknown multivariate dataset for supervised ML, where neither the characteristics of the dataset are known, nor are there any labels or labelled records.

Sometimes, the cost of labelling a dataset is significantly higher than the cost of creating it [Bernard, Hutter et al. 2018] and effective labelling solutions are still scarce. Analysts are confronted with the problem of making sense of a dataset, for example by identifying data characteristics such as frequent

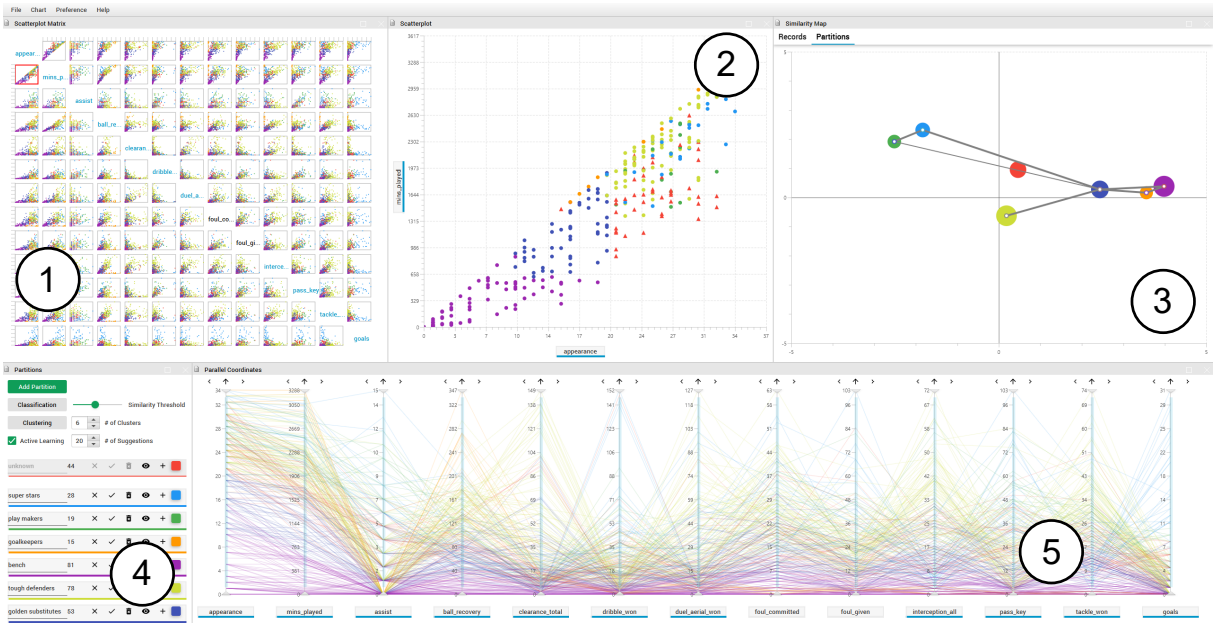


Figure 4.1: The scatterplot matrix (SPLOM) view ① shows the bivariate relationships between dimensions. The analyst can select a scatterplot from the SPLOM to show it in detail ②. The partition similarity map ③ shows partitions grouped by similarity and colour-coded as indicated in the partitions panel ④. If two partitions have associated dimensions (through user interaction), they are connected by a line. The parallel coordinates view ⑤ shows the dimensions of the dataset. Dimensions participating in the machine learning algorithms are indicated with a blue ribbon.

patterns or outliers. Active learning (AL) techniques, where the system periodically asks the user to label chosen records, can assist in the labelling process. However, since no labels exist at the beginning, AL techniques often suffer from bootstrap problems [Attenberg and Provost 2011].

Adding to the challenge is that an appropriate *label alphabet*, the vocabulary of labels, is generally unknown at the start of such a process, given an unknown dataset and/or users with ill-defined information needs. In some situations, different label alphabets might be appropriate, depending on the task at hand or a user's individual preferences. Analysts often derive the labels appropriate for a specific dataset and task from the data itself, exploiting the characteristics encoded in the multivariate data records and dimensions. In other situations, analysts rely on special domain knowledge to come up with initial labels. In any of these cases, neither AL tools nor the results of classifiers are particularly helpful for the determination of a label alphabet. Furthermore, the label alphabet is often subject to change during the labelling process itself.

Combining the strengths of humans and computers has been shown to be highly beneficial for the ML process [Amershi et al. 2014] as well as for information visualisation and visual analytics (VA) [Sacha, Sedlmair et al. 2017]. The visual interactive labelling (VIAL) technique [Bernard, Zeppelzauer, Sedlmair et al. 2018] combines ML principles with interactive visual interfaces for the effective selection of records for labelling. This principle has been adopted here. With the highly iterative VIAL process, a classifier can be continuously updated according to new label information provided by the user. Embedded AL strategies guide the user towards records which, once labelled, are likely to improve the underlying ML model. In mVis (**m**ultivariate **Vis**ualiser), this principle is complemented with interactive visual interfaces for data exploration, allowing the meaningful selection and labelling of records based on insights gained by the user, in addition to those suggested by AL. Figure 4.1 shows the user interface of mVis.

The interactive visual approach described in this work enables analysts to label records and create

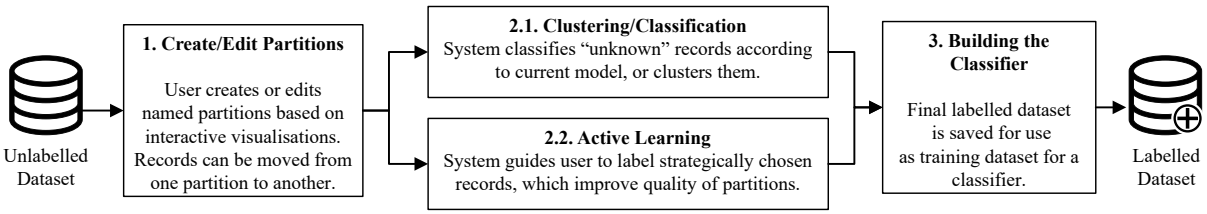


Figure 4.2: The workflow for interactive labelling. First, the analyst creates and names (labels) partitions in the dataset and assigns records to them. In the second and the third step, with guidance from the system, partitions are refined, and more records are added (labelled). After sufficient iterations, based on the quality of the result, the analyst saves the labelled dataset to be used as a training dataset for a classifier.

partitions of a previously unknown dataset in an effective and efficient way. While analysts may start without any knowledge about the dataset and the label alphabet, the output of the implemented approach is a labelled training dataset which can be used for supervised ML. The labelling process represents a pathway from unsupervised ML, through semi-supervised ML, to supervised ML. This pathway is guided by algorithms built upon both unsupervised and supervised ML principles. The approach presented here has three main components: (a) visual exploration, (b) interactive visual labelling, and (c) automatic guidance.

Firstly, the dataset can be explored interactively using a palette of linked visualisations, including scatterplots, a SPLOM, similarity maps, and parallel coordinates. These tools allow interactive visual exploration of a dataset's records and dimensions to both discover and then interactively label groupings, patterns, and outliers. Moreover, a novel view called the partition similarity map shows the similarity of partitions (each represented by a coloured node), based on the centroid of each partition. A link is drawn between two partitions if both partitions are associated with at least one common dimension. A dimension is associated with a partition, if the user interacted with that dimension while adding records to the partition.

Secondly, records can be selected and labelled in any of the interactive views, leading to labelled datasets which can be used for supervised ML. During the labelling process, dimensions that the user interacted with to perform labelling are added to the label as metadata. This solution facilitates labelling without the need for domain-specific visual representations by leveraging the structural information provided within a multivariate dataset, such as patterns and relations between records and dimensions. The original VIAL process is extended by incorporating classic k-means and hierarchical clustering to the supervised ML techniques.

Thirdly, clustering, active learning, and classifier algorithms are all available to support the effective and efficient selection of candidate records for labelling. In addition, using a new *automatic dimension selection* technique, interactions of the user with specific data dimensions are remembered and fed into the semi-supervised and supervised ML techniques. For example, if the user selected records in a scatterplot of dimensions A and B, and added these records to a partition, then dimensions A and B are associated with that partition. Initially, dimensions which are not interacted with play no role in the ML algorithms, but the user has final control over which dimensions should be included in or excluded from the ML algorithms.

4.2 Research Questions and Hypothesis

This chapter addresses RQ2: *How to use VA for building an ML model?* More specifically, how to use VA to effectively label a multivariate dataset. Therefore the following research question is asked.

Research Question 4.1 (RQ4.1): *How to use VA together with traditional ML techniques for interactive labelling of a multivariate dataset?*

The primary contribution of this chapter is to elaborate how linked interactive visualisations can be effectively integrated with classic ML algorithms to provide guidance during the labelling process without overwhelming the user. This work adds to explorations of the potentially large design space of visual analytics methods facilitated by active learning, and sets examples upon which to build future work.

A hypothesis correspondent to RQ4.1 is formed and tested in this chapter. To demonstrate the effectiveness of the approach, it has been incorporated into the mVis system and tested with a real-world football dataset.

Hypothesis 2 (H2): *By using interactive visualisation techniques, an analyst can build a machine learning model for a multivariate dataset.*

4.3 Interactive Visual Labelling

It is often the case that an analyst is confronted by an exploratory scenario in which the records in the dataset are unknown, and no labels are assigned to them. For ML applications, similar records must be grouped together and manually labelled in order to use the dataset as a training dataset. Since the definition of similarity varies from dataset to dataset, it is necessary to offer support to analysts to interactively group and label records and iteratively construct the label alphabet (L).

In an exploratory scenario, there is no single absolute L for a dataset. Based on the knowledge of the expert, L and the records assigned to each partition may vary significantly. Thus, a dynamic L is necessary to empower the analyst to build an appropriately labelled dataset fitting the purpose of the desired classifier. This includes allowing the analyst to (1) add new labels to L, (2) delete labels from L, (3) add or remove records to a label in L and (4) rename a label in L.

A partition, identified by P_i, is a set of records from the dataset, whereby each record must belong to one and only one partition. The union of all partitions P contains all records in the dataset. Each partition also has a label, l_i, which is a text string belonging to the label alphabet L, and a set of related dimensions Dim_i:

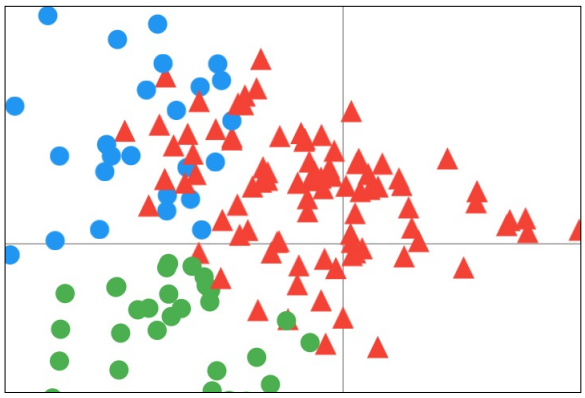
$$P_i = (l_i, Rec_i, Dim_i) \tag{4.1}$$

where:

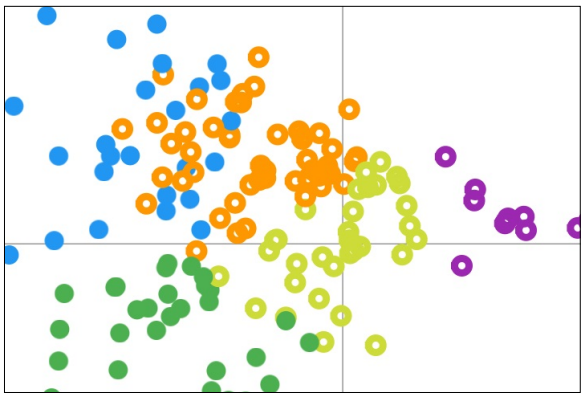
- l_i is one of the labels in the alphabet L. One label exists for each partition, one partition exists for each label.
- Rec_i is the set of all records labelled as l_i. There is a non-injective non-surjective function which maps records to partitions. In other words, every record is mapped to one and only one label at a time; $f : P \rightarrow L$, where f is the function which maps records to labels. The mapping is guided by the system, but is the analyst's task.
- Dim_i is a set of dimensions that the user interacted with while adding records to P_i. It is possible for a dimension to be associated with more than one partition, and there could be dimensions which are not associated with any partition.

4.3.1 Analyst Role: Selection and Labelling

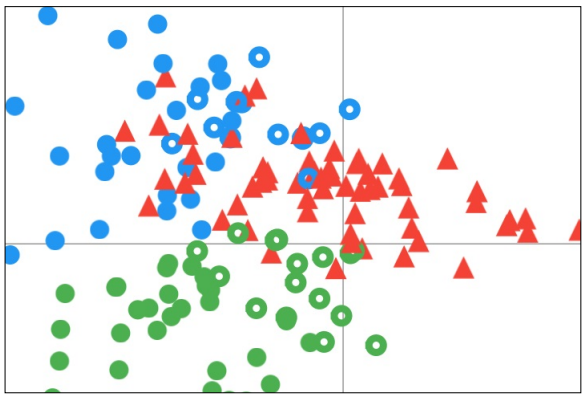
Figure 4.2 illustrates the workflow in which an analyst creates and edits partitions and labels records interactively. Initially, all records are assigned to a special partition labelled as *unknown*. In the first step, the analyst creates at least one partition, assigns records to it, and gives it a label. Later, the analyst can perform clustering and classification to label further records currently labelled as *unknown*. In the case of



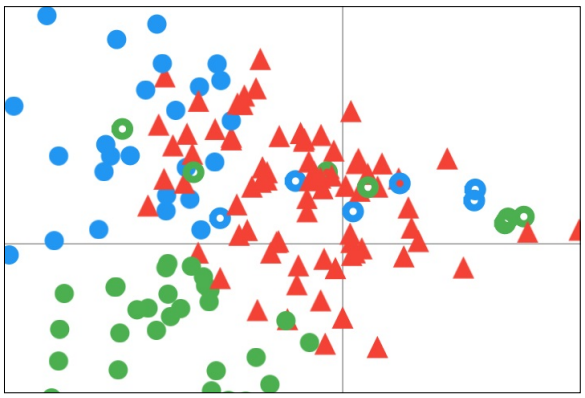
(a) The initial state with three partitions: green circles, blue circles, and red triangles (unknown).



(b) After clustering from (a), with k-means (k=3). The newly suggested clusters (partitions) are orange, yellow, and purple.



(c) After classification from (a) with a similarity threshold of 70%. Hollow circles are the system suggestions.



(d) After active learning from (a) in which the system suggested 30 records for labelling by the user.

Figure 4.3: The results of clustering, classification, and active learning in mVis, each applied to the initial state shown in (a). In each case, hollow circles indicate records with labels suggested by the system. Solid circles indicate previously approved labels. Solid red triangles indicate currently unlabelled records belonging to the unknown partition.

clustering, the system creates new partitions of *unknown* records and assigns temporary labels to them. In the case of classification, currently labelled records are used as a training set to label other *unknown* records based on existing partitions, which then potentially expands them. In either case, the system provides guidance by suggesting new labelled records, which the analyst can then approve or reject.

Periodically, the system suggests that the analyst should manually label a specific number of records by running active learning techniques. These records are wisely chosen to further resolve ambiguity in the dataset. The analyst investigates the result and decides if the alphabet and labels on records need further improvement. The process finishes when the analyst is satisfied with the quality of the result. The result of this process is a label alphabet (L) and a set of labelled partitions (P_i), in other words a labelled training dataset for a classifier. Records still labelled *unknown* may or may not be included in the output.

4.3.2 System Role: Guidance

The system's role is to suggest records for labelling to the analyst by visual clustering, classic clustering, classification, and active learning. Table 2.1 differentiates between these four kinds of technique.

In terms of visual clustering, the system provides similarity maps using one of three different projections: PCA, MDS, and t-SNE. Similar records are grouped by proximity and the analyst can efficiently create and modify partitions by visually inspecting these views.

In terms of classic clustering, the user can ask the system to cluster currently unlabelled records, using either k-means or hierarchical clustering. This results in a number of newly created partitions (i.e. clusters) with temporary labels, which the analyst can then either rename, approve, or reject.

Once sufficient numbers of records have been labelled, the analyst can use classification to help label further records. After performing the classification, the system calculates the similarity of each record (r_j) to each partition (P_i). The sum of all these scores for each record is always 100. The user can then define a *similarity threshold*. The system will suggest adding records with a similarity score higher than the threshold to the corresponding partition. If multiple partitions have a higher similarity score than the threshold, the system will choose the partition with the highest score. The user can either approve or reject the new suggestions. In classification, no new partitions or labels are created, but records may be added to the existing partitions P_i.

For active learning (AL), the system also requires a sufficient number of labelled records. It then chooses those unlabelled records which are most likely to further resolve ambiguity in the dataset, and asks the analyst to manually label them. Unlike clustering and classification, AL is not triggered by the user, but periodically by the system. Figure 4.3 shows the differing results of clustering, classification, and active learning in mVis.

The set of all dimensions associated (by user interaction) to at least one partition, *Dim*, is the union of all Dim_i. The above techniques do not always incorporate all of the dataset's dimensions in their various calculations. Instead, a set of *participating* dimensions is maintained by the system. Initially, the set of participating dimensions is set to be *Dim*, a feature called *automatic dimension selection*. However, the analyst has final control, and can include or exclude any dimensions from the set of participating dimensions. The final result of the workflow is a labelled dataset which includes P_i, L, and a set of related dimensions.

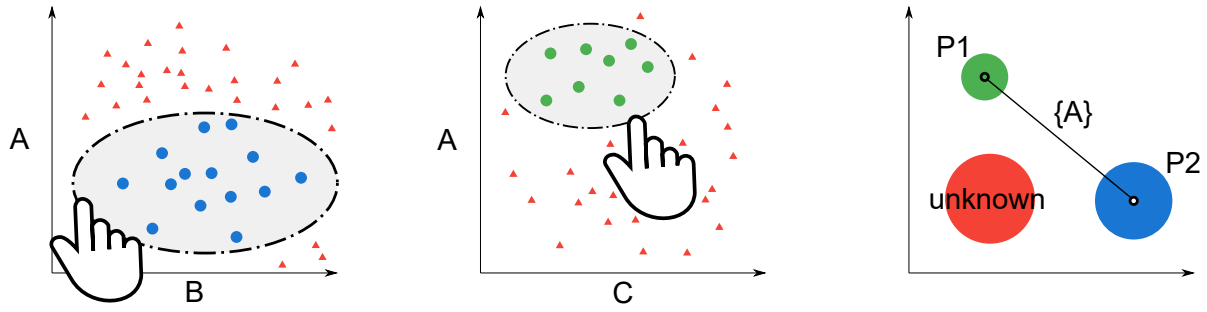
4.4 mVis System Overview

The mVis system consists of four data visualisation views and a panel to control partitions. mVis is written in Java and uses JavaFX for its user interface. It supports traditional mouse and keyboard as well as multi-touch user input. The system has been tested on a PC with a 3.4 GHz Intel i7-6700 CPU and 64 GB of RAM, running 64-bit Windows 10.

4.4.1 Visualisations and Partitions Panel

The four linked exploratory data visualisations built into mVis are: SPLOM, scatterplot, similarity map (projection by PCA, MDS, and t-SNE), and parallel coordinates plot. All the visualisations are connected through standard brushing and linking, so selections and changes in one view are reflected in all other views. Moreover, the user can close, rearrange, or enlarge any view. Axis tick labels in the scatterplot and parallel coordinates views reflect the original values in the dataset. Coordinates in the SPLOM view are normalised, so axis tick labels are omitted.

The SPLOM provides an overview of the entire dataset by showing all bivariate projections of n dimensions. The result is a matrix of n^2 scatterplots [M. A. A. Cox and T. F. Cox 2008]. The SPLOM



(a) Creating P_1. **(b)** Creating P_2. **(c)** Similarity map.

Figure 4.4: Records are added to partition P_1 (blue) from AB, then to partition P_2 (green) from AC. The partition similarity map shows a link between P_1 and P_2 because they are both associated with dimension A.

can indicate both patterns of records in two dimensions and correlations between pairs of dimensions, which can then be examined in individual scatterplots.

Individual scatterplots are widely used for regression analysis [Shao, Mahajan et al. 2017] or exploration of local patterns (see Chapter 3). In mVis, the user can select a scatterplot in the SPLOM, which is then shown enlarged in the scatterplot view.

In mVis, the parallel coordinates view supports several interactions, including brushing and selection of records, filtering of records by dragging sliders at the top and bottom of each axis, reordering axes, and inverting axes.

The similarity map view provides two kinds of similarity map: a similarity map of records and a similarity map of partitions. The record similarity map shows all the records in the dataset visually clustered by similarity, using one of three projection techniques: PCA, MDS, or t-SNE. More similar records are closer together in the similarity map. The default projection technique is t-SNE, but the user can choose a different technique in the preference menu.

The partition similarity map shows all currently defined partitions, grouped by similarity in the form of a node-link diagram. Each partition is represented as a circular node, whose size corresponds to the number of records in the partition. If two partitions share associated dimensions, then a line (link) is drawn to connect them, whose width corresponds to the number of shared associated dimensions. Figure 4.4 illustrates how such a diagram is created. First, in Figure 4.4a, the analyst creates a partition P_1 containing records selected in the scatterplot of dimension A against dimension B (AB). Later, in Figure 4.4b, the analyst assigns records to P_2 from the scatterplot AC. Since both partitions are associated with dimension A, there a link is drawn between P_1 and P_2, as shown in Figure 4.4c.

The partitions panel shown in Figure 4.5 gives the analyst the possibility to create new partitions, assign records to partitions, and delete partitions. The name (label) of a partition can be edited and the colour assigned to it can be changed. A special partition labelled *unknown* contains all currently unlabelled records and is initially coloured red. If a partition is deleted, all records contained within it are returned to the *unknown* partition. The analyst can temporarily hide the records in a given partition. Clicking the "+" button next to a partition adds currently selected records to it.

Records which have been manually assigned to a partition or approved by the analyst are considered to be "ground truth" and are represented by solid circles in the SPLOM, scatterplot, and record similarity map. Hollow circles represent records with a suggested partition, colour-coded according to the partition. Unlabelled records belong to the *unknown* partition and are represented by solid triangles, in the colour assigned to the *unknown* partition (initially red, but the colour can be changed by the analyst).

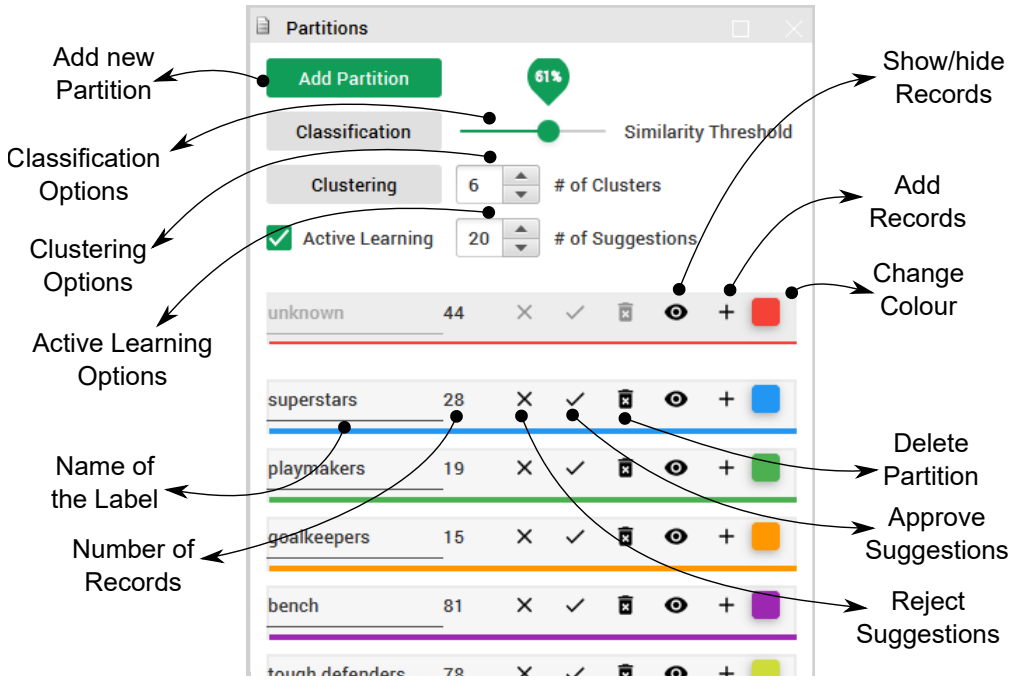


Figure 4.5: The partitions panel. In the upper part of the panel, the analyst can create partitions and obtain suggestions for records to add to them. The lower part of the panel is for manipulating existing partitions.

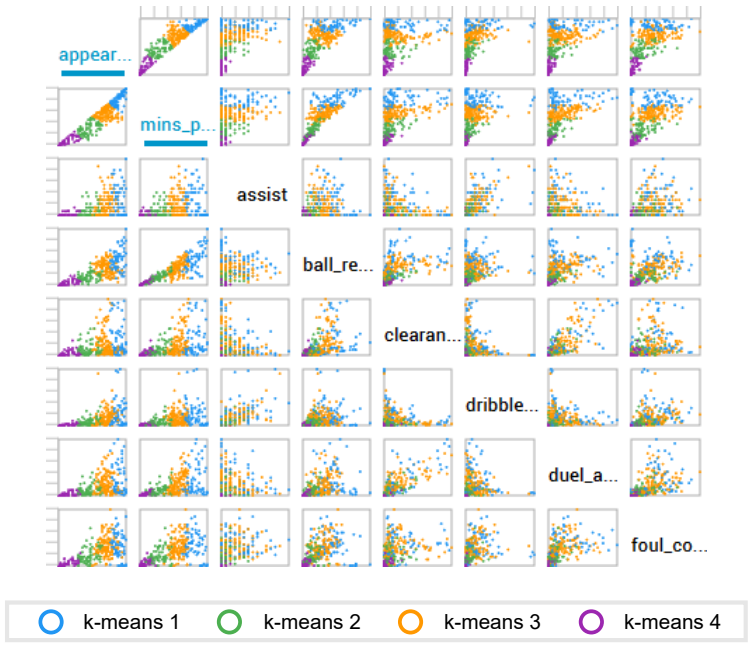
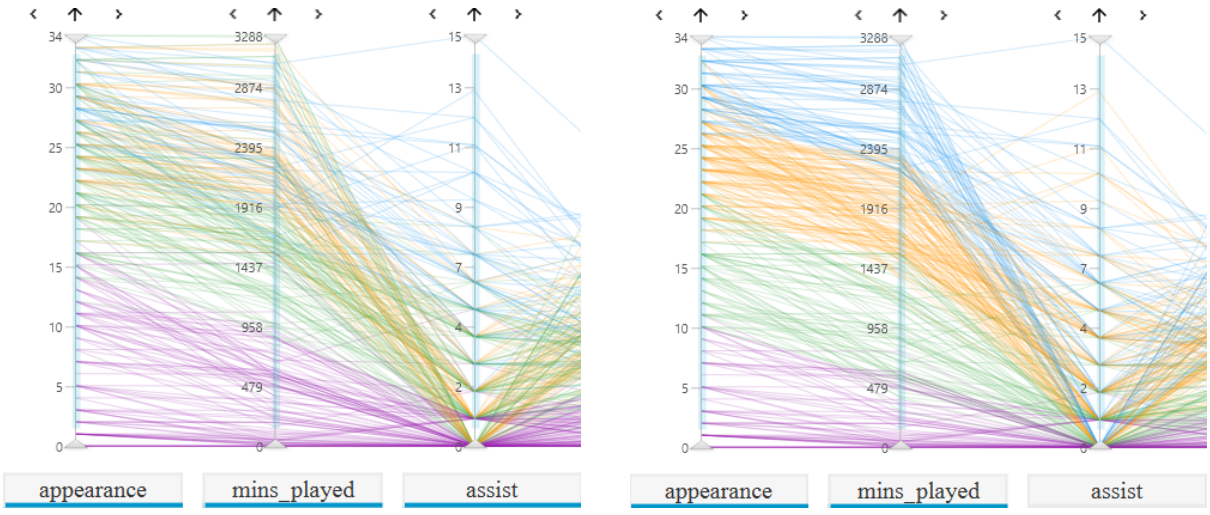


Figure 4.6: The SPLOM after k-means clustering (k=4) with automatic dimension selection. A blue ribbon beneath a dimension name indicates its participation in the ML technique. The first two dimensions appearances and mins_played from the football dataset have participated in the clustering, which is reflected in the better results in their rows and columns.



(a) Three participating dimensions.

(b) Two participating dimensions.

Figure 4.7: Part of the parallel coordinates plot after hierarchical clustering (k=4). The clusters are more visually appealing in (b).

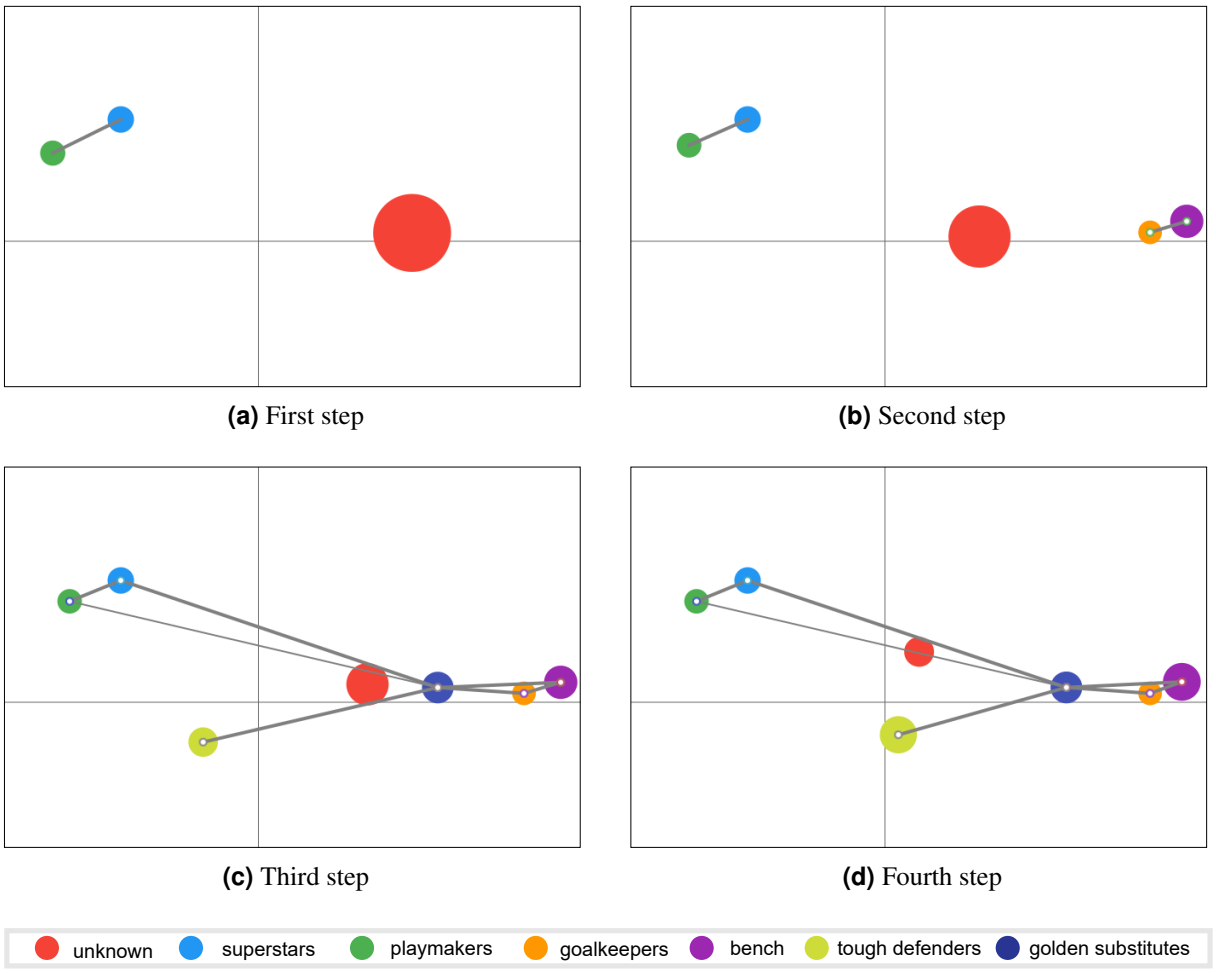
In the upper part of the partitions panel, the analyst can initiate ML techniques such as clustering and classification to obtain suggestions for records to assign to partitions. Such records become hollow circles and are recoloured to the suggested partition's colour until either approved or rejected by the analyst by clicking the Reject or Approve buttons next to each partition in the panel. Suggested records which are rejected become solid (red) triangles again and are moved back into the *unknown* partition. Approved records become part of the partition and are henceforth represented by solid circles.

4.4.2 Machine Learning Modules

Various ML algorithms are implemented to support the interactive labelling process, including dimensionality reduction, clustering, classification, and active learning. All of these algorithms are implemented using the Java library called DMandML [*DMandML* 2018]. Interactions with an ML algorithm can be unintuitive and overwhelming to use at times. mVis uses simple widgets and a minimal number of exposed parameters to keep interactions intuitive.

While assigning records to partitions, the system keeps track of the dimensions the user interacted with, maintaining a set of associated dimensions for each partition. By default, only those dimensions associated with at least one partition participate in the ML algorithms. The user can toggle participation of a dimension by clicking on the dimension name in the SPLOM or parallel coordinates view. Participating dimensions are indicated by a blue ribbon beneath the dimension name. Figure 4.6 shows k-means clustering (k=4) utilising only two of the eight available dimensions. Figure 4.7 demonstrates the effectiveness of automatic dimension selection when hierarchical clustering is performed on the dataset.

At any stage, the analyst can perform clustering by clicking on the clustering button in the partitions panel. The system will then cluster all currently unlabelled (*unknown*) or unapproved records using k-means or hierarchical clustering. By default, mVis uses k-means, but the user can change the algorithm by selecting hierarchical in the menu. For each cluster, a new partition is created and given a temporary name (label) of the form `k-means #cn` or `hierarchical #cn`, where `#cn` is the number of the cluster. Records assigned to a cluster are simply suggestions by the system and require subsequent user approval.



(a) First step

(b) Second step

(c) Third step

(d) Fourth step

Figure 4.8: Four steps of labelling the football dataset, shown in the partition similarity map. (a) The user manually creates superstars and playmakers partitions. (b) After a clustering step using k-means, two partitions called goalkeepers and bench are approved by the user. (c) The user creates tough defenders and golden substitutes partitions and assigns records to them. (d) The user performs active learning to label more records. The final result is a label alphabet with seven members.

Alternatively, once sufficient records have been assigned labels, the analyst can run a classifier to classify those records which are currently either *unknown* or unapproved. The system then runs a *Random Forest* classifier using the already labelled (approved) records as a training set. The user can control the number of suggestions by adjusting the similarity threshold with the slider next to the Classification button. While the slider is adjusted, a number indicates its precise value. With a higher threshold, only those records more similar to a specific partition will be suggested. Similar to clustering, the analyst can then approve or reject the classification result.

Periodically, the system actively guides the user to manually label a number of records using active learning. The suggested labels can either be approved or rejected. The number of suggested records can be fine-tuned and active learning can be turned off completely with the checkbox in the partitions panel.

The current design of mVis has visualisation and algorithmic limitations. Regarding the visual scalability of the label alphabet (number of partitions), upto around twelve distinct colours can be comfortably distinguished [Harrower and Brewer 2003]. The SPLOM and parallel coordinates views are limited by the amount of available screen space. mVis runs in real-time with a football dataset comprising 42 dimensions and 318 records on a 25-inch desktop display at a resolution of 2560×1440. One possibility

to increase scalability would be to apply subspace clustering to provide an initial set of records and dimensions to explore [Hund et al. 2016]. The currently implemented ML algorithms run in real-time for the aforementioned number of partitions and dimensions.

4.5 Football Dataset Use Case

To test Hypothesis 2, the following use case that utilises a football dataset of players from 16 clubs participating in five top European leagues in the 2017/18 season [Berger et al. 2018] is introduced. The records are individual players, the dimensions are players' attributes such as the number of match appearances, committed fouls, assists, pass accuracy, and so forth. The dataset comprises 318 records and 13 dimensions.

The goal of the analyst exploring this dataset is (1) to group the players into labelled partitions based on their characteristics, and (2) to use the dataset to train a classifier for other seasons of the same or even entirely different football leagues.

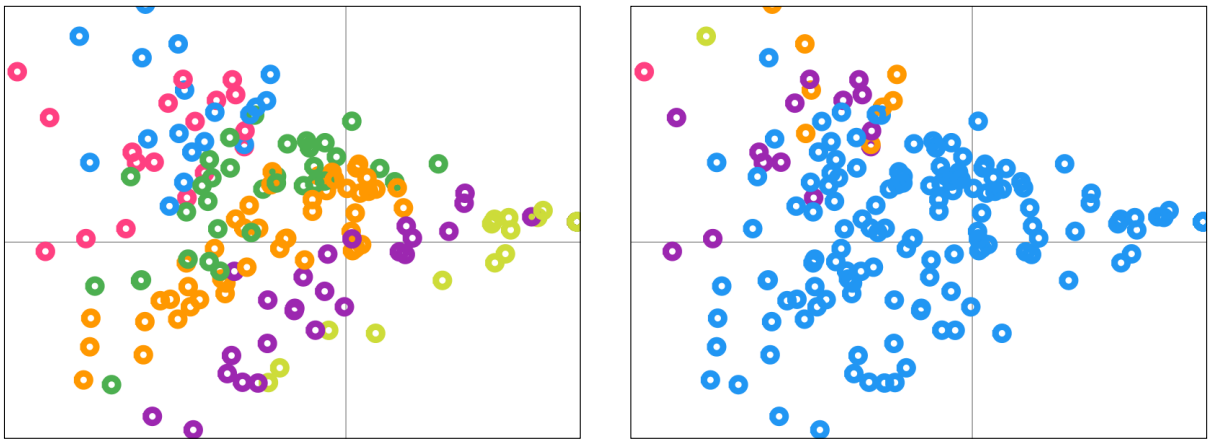
For an initial grouping, the analyst wants to identify match-winning players and label them as `superstars`. The analyst proceeds by selecting the scatterplot of goals against assists in the SPLOM. The analyst creates a partition, labels it `superstars`, and includes all data records with high numbers of goals and assists.

Another important category of players are the so-called `playmakers`, having a high number of assists and key_passes. By filtering players with a high number of assists and key_passes in the parallel coordinates view, the analyst can find records to add to the `playmakers` partition. To expand the label content so that not only top players are included, the analyst searches for players similar to those selected. To this end, the analyst sets the *Classification Threshold* slider in the partitions panel (see Figure 4.5) to 60% and clicks the Classification button. As a result, the system suggests 100 records be labelled as `playmakers` and 20 as `superstars`. The analyst realises that this is a large number of players to be added to each partition and decides to reject the suggestion. Later, the analyst performs another classification with the slider at 80%. This time, 15 records are suggested to be added to `playmakers` and 5 to `superstars`. The analyst accepts the suggestion by clicking the Approve button of both partitions. The partition similarity map in Figure 4.8a shows the state of the dataset after creating the partitions `superstars` and `playmakers`.

Apart from these two obvious choices, the relationships between other dimensions are unfamiliar to the analyst. The analyst turns off the *automatic dimension selection* feature, chooses 4 as the value in the *# of Clusters* field, and performs a k-means clustering. By making all partitions except one invisible, the analyst inspects the newly suggested partitions one by one. The first suggested partition is `k-means 1`, containing 16 records. The analyst realises all the dimensions for these records are zero except appearance, mins_played, and ball_recovery. Therefore, the analyst renames the `k-means 1` partition to `goalkeepers`. Similarly, the analyst renames `k-means 2` with 88 records to `offensive players`. This partition is associated with the dimensions key_passes, dribbles_won, and goals. Next, the analyst renames `k-means 3` with 71 players to `defensive players`, since it is associated with ball_recovery, clearances, aerial_duels_won, fouls_committed, and interceptions. Finally, the partition `k-means 4` with 116 records is renamed `bench`. This partition is associated with a low number of appearances and mins_played.

The goal is not to create partitions based solely on a player's role on the field, so the analyst decides to delete the partitions `offensive players` and `defensive players` by clicking their Delete buttons, but to retain the partitions `goalkeepers` and `bench` by clicking their Approve buttons. Figure 4.8b shows the state of the dataset after this step.

Similar to the group of match-winning `superstars`, the analyst wants a label for defensive players having a high impact on the team. From the previous exploration, the analyst already knows which dimensions are associated with defensive characteristics. Therefore, the analyst creates the `tough defenders`



(a) K-means. **(b)** Hiearchical.

Figure 4.9: The results of k-means and hierarchical clustering for k=6, using offensive attributes of football players.

partition characterised by their performance in the dimensions aerial_duels_won, interceptions, and tackles_won.

Exploring further, the analyst selects all records which (1) belong to the `bench` partition and (2) have either a high number of goals, key_passes, clearances, dribbles_ won, assists, or aerial_duels_won and calls the new partition `golden substitutes`. To further support the analyst, the remaining unlabelled records (belonging to the *unknown* partition) can be suggested to existing partitions via active learning. This helps refine existing labels and increasing the overall quality, an option which is not possible in traditional ML techniques.

The analyst investigates the result shown in the partition similarity map of Figure 4.8d. The `tough defenders` partition is linked to `golden substitutes` partition, since they are both associated with the clearances dimension. Also, `playmakers` and `superstars` are relatively close to one other in the partition similarity map, possibly because `playmakers` and `superstars` share similar offensive characteristics. Since the user interacted with eleven dimensions, only two dimensions are not highlighted with a blue ribbon.

The result of the session is a labelled football players dataset with meaningful partitions, which can be used as a training dataset for a classifier for other seasons or different leagues.

4.6 Pre-Studies for mVis

There are many ways to evaluate interactive systems for visual analysis [K. Andrews 2006; K. Andrews 2008] and many motivations behind such evaluations [Lam et al. 2012]. However, as previous researchers have noted, it can be challenging to evaluate such systems [Plaisant 2004; Carpendale 2008; Crisan and Elliott 2018]. Datasets can vary wildly and tasks are often dependent on the kind of data being explored. In many applications, domain experts are recruited for evaluation. However, a domain expert is not always available or willing. It is also hard to measure and compare the "insights" which such systems are designed to discover [North 2006]. The difficulty of running controlled experiments has lead to the increasing use of qualitative evaluation methods involving case studies and observation of individual users [Shneiderman and Plaisant 2006; Perer and Shneiderman 2009].

To further test mVis, and Hypothesis 2, in this section, a pre-study evaluation of the mVis system, comprising two case studies each in a different domain (collaborative intelligence and daily activities) is described. In each case study, a volunteer researcher with no previous experience of mVis was observed

Figure 4.10: The facilitator (left) and researcher (right) conducting a case study with mVis introduced in Chapter 4.

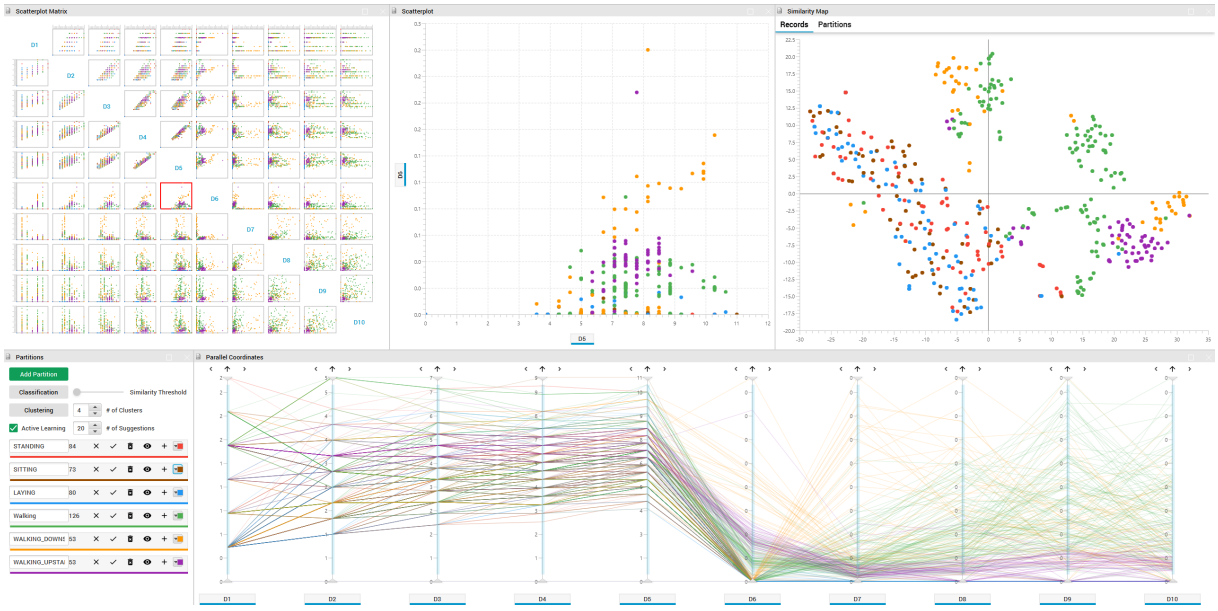


Figure 4.11: A screenshot of mVis taken during the second case study.

as they used mVis to explore, label, and verify a dataset from their own domain. The researchers were asked to talk out loud as they worked to provide greater insight into their thought process [Ericsson and Simon 1984]. Afterwards, the researchers participated in a semi-structured interview. This type of evaluation is useful to (1) test the usability of the system, (2) understand how the current implementation helps the analyst with their tasks, and (3) identify important missing features. The results will be used to inform and plan future evaluations. Figure 4.10 shows the setup of a case study. Figure 4.11 shows a screenshot of mVis during the second case study.

4.7 Pre-Studies Methods

Since the focus of mVis is not on a specific domain, gathering requirements and evaluation feedback is complex. It is necessary to conduct several domain expert studies to identify common requirements and improve usability of the system. Since the system is still evolving, qualitative studies involving observation and thinking-aloud, followed by semi-structured interviews are preferred over other types of user study.

The pre-study evaluation of the mVis system comprised two case studies, each with a domain expert. In each case study, the domain expert was a volunteer researcher with no previous experience of mVis. The faciliator first provided the researcher with a five-minute introduction to mVis. Then, the researcher was observed as they used mVis in their own office environment to explore, label, and verify a familiar dataset from their own domain, while thinking out loud [Ericsson and Simon 1984]. The facilitator sat next to the participant and took notes, and provided assistance with mVis when asked. Afterwards, the participant was interviewed in a semi-structured way to discover (1) their general impression of mVis (2) any missing features, and (3) in which stages of analysis mVis proved useful.

4.7.1 Case Study 1: Collaborative Intelligence Dataset

The volunteer researcher in the first case study was Monika. She is working on a collaborative intelligence platform dataset consisting of 718 records and ten dimensions. Each record represents a user and the dimensions are quantitative numbers associated with activities of the user on the platform (for example, number of *comments* and number of *reports*). Monika has been working with this dataset for over a year.

The session was conducted on a laptop with a 12-inch display in Monika's office. The facilitator imported the dataset into mVis after cleaning it. The facilitator then explained the user interface of mVis for around 5 minutes and asked Monika to freely explore the dataset while verbalising her thoughts. She started by investigating patterns in pairwise dimensions. For example, she realised that users who participate in *chats* do not often give *comments*. She also identified a relationship between *comments* and *reports*. Later, she performed several brushing interactions, using either the parallel coordinates plot or scatterplot. After the initial exploration phase, she created two partitions using first k-means and then hierarchical clustering. The new partitions mainly separated active and inactive users. She deleted all partitions and then performed another k-means clustering with k equal to four. Monika removed *votes* from the participating dimensions, since it is not a good indicator for clustering based on her prior experience. The result was better this time. For example, she identified that one of the clusters are users having a low number of *reports*. In summary, Monika used mVis primarily to (1) find new patterns among users and dimensions, (2) verify her previous observations about the dataset, and (3) create meaningful partitions.

A semi-structured interview was conducted shortly after the observation phase. The facilitator started by asking Monika's impression of mVis. She said she could confirm many previous observations in her explorations and even find new ones. She stated "Understanding this dataset would have been so much easier, if I had had this tool a year ago". The facilitator asked which features mVis is missing. She

noted a lack of interactive help, lack of an easy-to-use dataset importer, lack of a feature to save previous partitions, and no means to visually compare old and new partitions. Monika later added that mVis is especially useful for people who do not have enough knowledge to use other tools such as R and Python. Finally, the facilitator asked in which stages of analysis mVis can be useful. Monika mentioned mVis is useful in the initial phase of data exploration to understand dimensions and records before creating advanced ML models. The duration of the session after importing the dataset was 55 minutes.

4.7.2 Case Study 2: Daily Activities Dataset

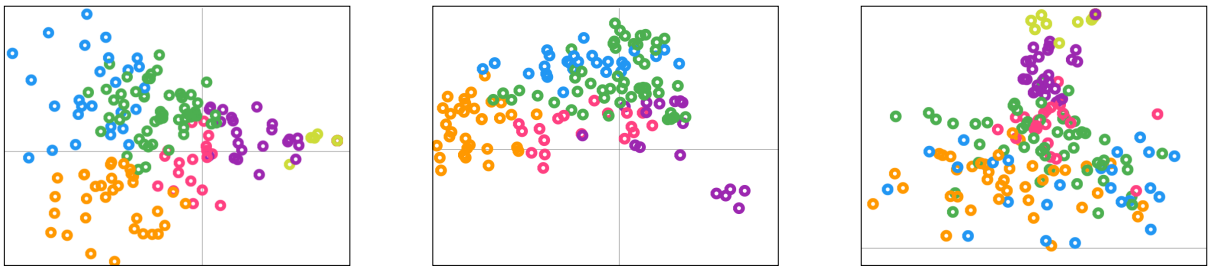
The participant in the second case study is Jian, a researcher working on a dataset about daily activities. The dataset has 25 dimensions and 412 records collected by feature extraction from time-series signals. The signals are gathered by placing various sensors in users' pockets to observe their daily activities. Each record belongs to a specific user and has already been manually assigned to (labelled with) exactly one of six states (classes) denoting daily activities: *walking*, *walking upstairs*, *walking downstairs*, *sitting*, *standing*, and *laying*. Figure 4.11 shows this dataset in mVis.

The session was conducted in Jian's office on a PC with a 24-inch display. The PC screen and user's voice were recorded on video. First, Jian explained the dataset to the facilitator and remarked that he has been exploring the dataset for six months using conventional data science tools such as R, Python, and SQL. He mentioned that he never tried to visualise the dataset using standard visualisation techniques. After introducing mVis to him, the facilitator imported the dataset into mVis. Jian reacted to the visualisation by stating "this looks great". Since the dataset was already labelled, he focused on finding relationships between dimensions and partitions. He first imported all 25 dimensions, and later decided to include only the first 10 dimensions into mVis. He started by identifying dimensions and their relationships with partitions. For example, he realised a dimension called *D8* can separate records labelled as *laying* from the others. By looking at the SPLOM, he discovered that many patterns recur among pairs of dimensions. He remarked that mVis does an excellent job in grouping dimensions and partitions. He later performed clustering to observe the differences between manual labelling and automatic partitioning using ML algorithms. At this point, he wished there was a visual comparison tool to compare previous and current partitions. In summary, Jian used mVis to (1) identify correlations between dimensions and partitions, (2) find relationships between pairs of dimensions, and (3) verify the manually labelled ML model in the dataset.

After the observation phase, the facilitator conducted a semi-structured interview. He first asked Jian about his impression of mVis. Jian mentioned that mVis might be useful for finding patterns in a dataset and he would use it for data exploration and ML model validation. Regarding missing features, he made two suggestions. First, a comparison tool to compare various partitions and secondly a guidance module to explain patterns in scatterplots. Lastly, he answered the question about which stage of data analysis mVis is useful by mentioning initial exploration. He added mVis could also play a crucial role in the validation phase. The duration of the session, including the introduction to mVis was 42 minutes.

4.8 Discussion

Characterising, comparing, and grouping (partitioning) the records in a dataset are among the most essential tasks in data analysis. The implemented approach supports these tasks with an interactive visual labelling tool. Using interactive visualisations, an analyst can identify and label groups of records in a dataset initially containing no pre-labelled records. Once the analyst has provided an initial labelling, the system supports labelling more records via clustering, classification, and active learning. With the help of clustering, the analyst can find structures in the dataset which may not be visible by manual exploration. Using classification, the labelled data will be used as a training set for records which are not yet labelled. Moreover, the active learning module regularly makes strategic suggestions to improve the quality of partitions. The user is always responsible for approving or rejecting suggestions, which increases overall



(a) MDS **(b)** PCA **(c)** t-SNE

Figure 4.12: The three projection techniques provided by the record similarity map. The colours were assigned by an initial k-means clustering with k=6.

trust in the result. As the presented use case shows, algorithmic support helps efficiently propagate current labelling to more records. The approach supports both the creation of a new label alphabet and the refinement of an existing label alphabet.

Currently, mVis supports both k-means and hierarchical clustering. Although k-means is more scalable and hierarchical is more flexible, neither is superior to the other. It is the responsibility of the domain expert to choose the most suitable algorithm in a specific situation. Figure 4.9 shows the results of k-means and hierarchical clustering in the football dataset.

Three projection algorithms (MDS, PCA, and t-SNE) are supported for the record similarity map. Research by Bernard, Hutter et al. [2018] shows that users prefer t-SNE as a dimensionality reduction technique for labelling tasks and later switch to PCA and MDS for validation. Therefore, the default algorithm in mVis is t-SNE. Figure 4.12 shows the differences between these algorithms, performed on the football dataset.

The pre-study with two case studies involved a combination of thinking-aloud, observation, and interview. It demonstrated the general utility of mVis and illuminated future directions. mVis is a general purpose system and is not designed for a specific domain, therefore it is crucial to work with a variety of analysts and domain experts to define common analysis approaches and goals and address these in the system. This pre-study, showed interactive visualisation techniques are a proper tool for building an ML model, and consequently proved Hypothesis 2.

One of the key observations of the study is how the nature of the dataset can change interactions with the system. For example, in the case of the collaborative intelligence dataset, the focus was on selecting individual records and investigating each dimension thoroughly. The smaller nature of the dataset allowed the analyst to do this. In case of the daily activities dataset, it was not possible for the analyst to explore all records and dimensions. The analyst was more interested in typical pairwise patterns and relationships between dimensions. By conducting more case studies on a variety of datasets from a variety of domains, it is hoped that further behavioural patterns and typical tasks will be revealed.

Chapter 5

Interactive Visual Labelling versus Active Learning

"The resemblance between reasoning and love;
Is like comparing a dew versus the ocean"

[Hafez, Persian poet, 1315-1390.]

It is shown that interactive visual labelling is an effective approach for labelling multivariate datasets. Therefore, the next question is, how this novel approach performs in terms of accuracy, compared to active learning techniques. By conducting a user study, this chapter proves that interactive visual labelling can outperform active learning approaches.

5.1 Introduction

Labelling is assigning a class from the label alphabet to an instance (a record) in a multivariate dataset. Supervised machine learning algorithms, such as classifiers [Bishop 2006], must be trained on a labelled dataset in order to perform. These methods learn how to generalise new data, based on existing known data examples which are provided with a class label. Creating a training dataset is essential to find a small subset of a dataset that delivers the best accuracy for the classifier. Although labelling a dataset is necessary, it can be a dull, time-consuming, and expensive task.

To address this problem, *active learning* algorithms can help the analyst by suggesting instances to label [Settles 2009]. Active learning algorithms effectively reduce the number of records which need to be interactively labelled. Active learning techniques require heuristics for record selection, which often depend on the classification problem or data characteristics. Furthermore, *interactive visual labelling* (VIAL) [Bernard, Zeppelzauer, Sedlmair et al. 2018] tools build explorable visual overviews on top of active learning algorithms and can outperform classic active learning techniques in terms of accuracy [Bernard, Hutter et al. 2018]. Such combined tools allow an analyst to label a multivariate dataset in a visual environment, while receiving feedback and guidance from the system. Based on the overall data characteristics perceived by the analyst, conscious choices can be made as to what distinguishes groups of data and how many groups there should be, and representative records can be labelled. Immediate feedback can be given regarding the current set of labelled records, for example by visualising changes and improvements to the given classifier in response to given changes in labelling. Thereby, users can also gain an understanding of which choices affect the classifiers, and hence contribute to understandable and explainable machine learning models.

Since there are multiple visualisation and interaction techniques, the following research question arises: *How do characteristics of these techniques and datasets affect performance and user experience*

Figure 5.1: The mVis tool, showing the SPLOM at top left, detailed scatterplot top middle, similarity map top right, and parallel coordinates bottom right, for the MNIST2 dataset. The partitions panel at bottom left shows the currently defined classes (label alphabet). Instances are colour-coded by class, here green for 1s and blue for 0s. Instances with confirmed labels are shown as crosses in the scatterplots and similarity map and as thick lines in the parallel coordinates. Suggestions from the classifier are shown as solid circles in the scatterplots and similarity map and as thin lines in the parallel coordinates.

for visual interactive labelling tasks? This key question will be broken down into several sub-questions in Section 5.3. To address them, this chapter describes a comparative user study of three well-known interactive visualisation techniques for visual labelling: *similarity map*, *scatterplot matrix* (SPLOM), and *parallel coordinates* [Inselberg 1985]. Using the existing mVis visual data exploration tool introduced in Chapter 4, nine machine learning experts labelled two multivariate datasets in each of these three views separately. The quantitative measures from these tasks are accumulated and compared to each other and to active learning algorithms. In addition, the techniques are compared to each other in terms of user experience. The results confirm that involving the user in labelling using visual exploration facilities can improve the machine learning process and enhance the ML model understanding.

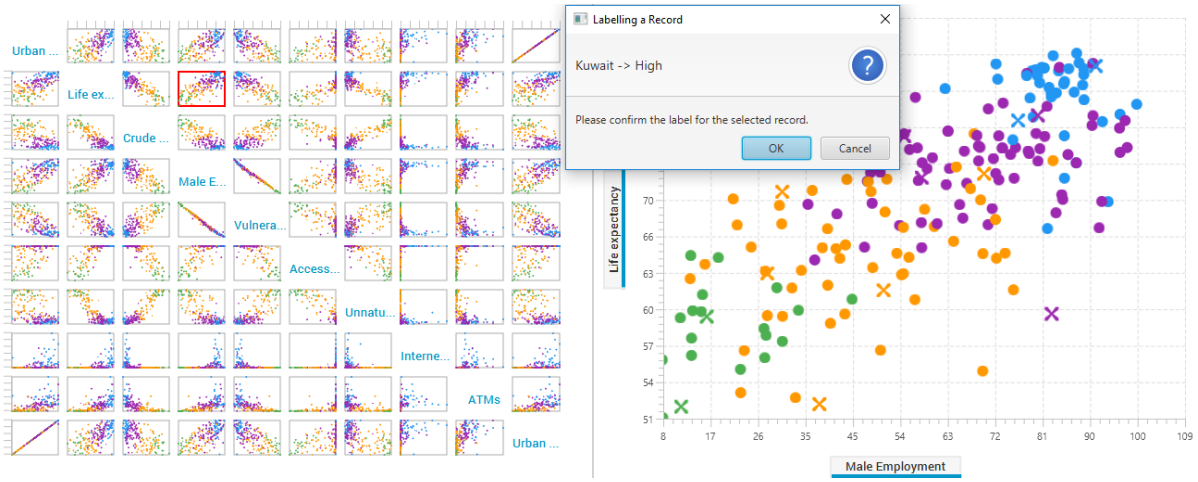


Figure 5.2: The SPLOM with scatterplot visualisation of the WB dataset, as used by a test participant. Instances are colour-coded by class. Instances with confirmed labels are shown as crosses, suggestions from the classifier are shown as solid circles. The user has selected the scatterplot of Life Expectancy versus Male Employment in the SPLOM on the left and has selected the instance of Kuwait for labelling in the detailed scatterplot view on the right. The dialogue on the upper middle of the screen asks the user to confirm the label for that instance.

5.2 Methods

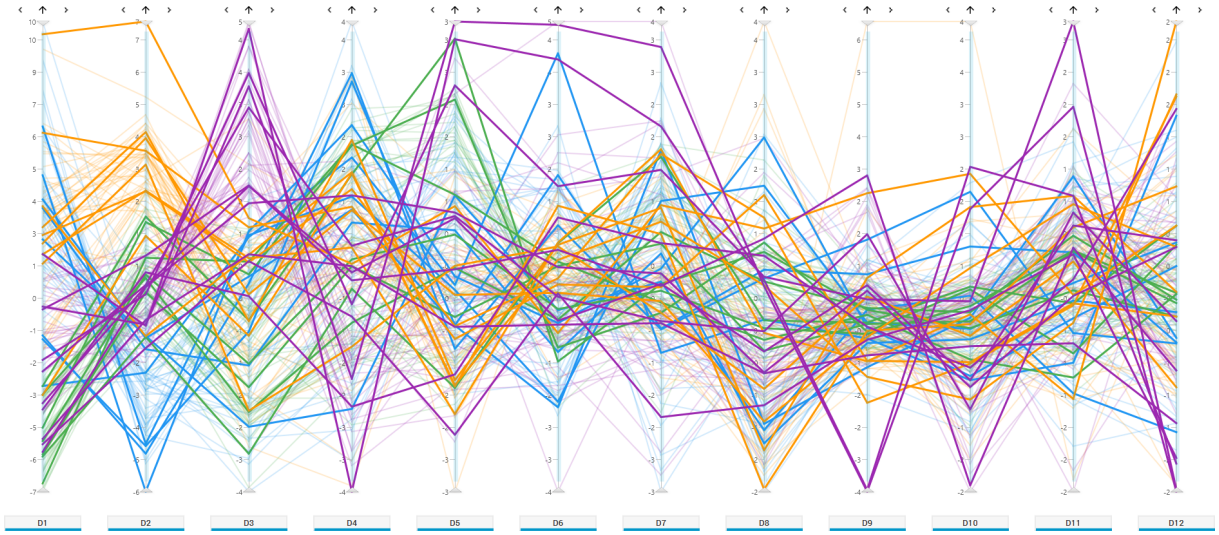
Using mVis, the performance of three different visualisation techniques for labelling a multivariate dataset was compared. Figure 5.1 shows mVis with a two-class subset of the MNIST dataset [LeCun et al. 1998]. Since prior studies have shown that users prefer t-SNE over PCA and MDS for interactive visual labelling, t-SNE [Maaten and Hinton 2008] algorithm is used for the similarity map.

For the SPLOM, all bivariate combination are shown in a matrix, and the user can select any of them to examine more closely in the scatterplot view. In the Parallel Coordinates view, the analyst can rearrange or invert dimensions and filter out records.

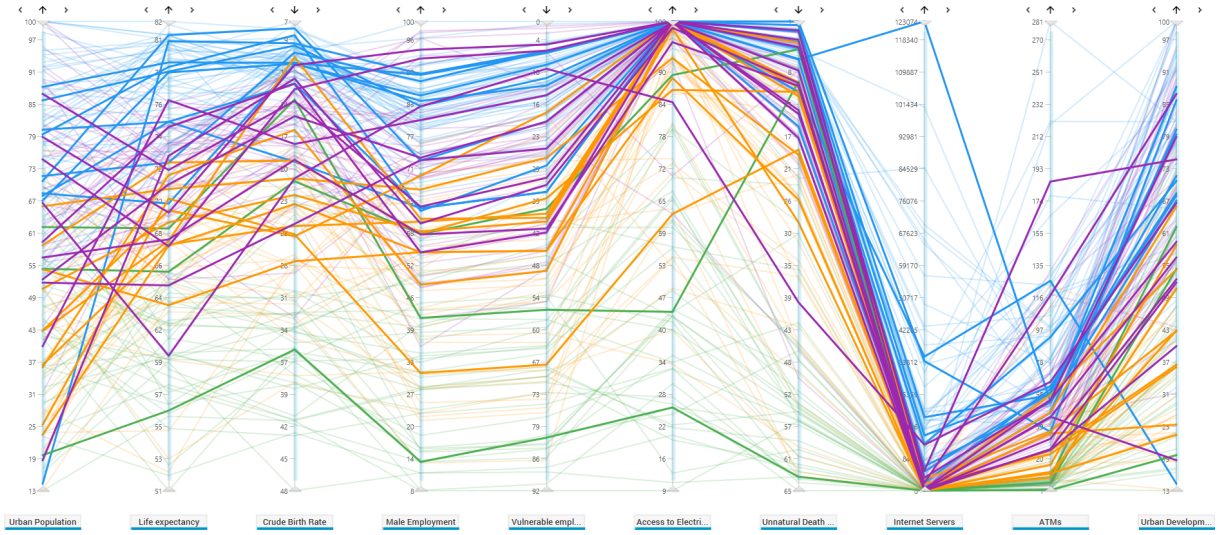
In general use, mVis allows the analyst to select one or multiple instances for labelling. Every time a set of instances is labelled, the Weka implementation of a Random Forest Classifier [Hall et al. 2009] runs in the background and suggests potential labels for all currently unlabelled instances by colour-coding according to their suggested class. Instances whose labels have been confirmed by the user are made visually distinct from instances with labels suggested by the classifier. Confirmed instances are shown as crosses in the scatterplots and similarity map and as thick lines in the parallel coordinates. Suggestions are shown as solid circles in the scatterplots and similarity map and as thin lines in the parallel coordinates. For the experiment described in this chapter, the user was restricted to selecting a single instance at each step, which was then assigned its pre-assigned class.

Later, in order to assess the classification performance of the interactive visual labelling techniques, three methods were used: active learning, greedy selection, and random selection. Three active learning methods were used: Smallest Margin, Entropy-Based Sampling, and Least Significant Confidence and the average accuracy in each step was used to compare the results. For the greedy method, the classifier was run for all possible instances for labelling, and the one with the best accuracy was selected. Greedy selection represents the best possible labelling result, and is the theoretical upper limit of what could be achieved by any visual labelling technique or active learning strategy. The random selection of instances was run 200 times, and the average accuracy in each step was used to compare the results. Random selection represents a practical lower limit for the accuracy a classifier should achieve.

The work of Bernard, Hutter et al. [2018] was chosen to describe the strategies of users for selecting



(a) MNIST4.



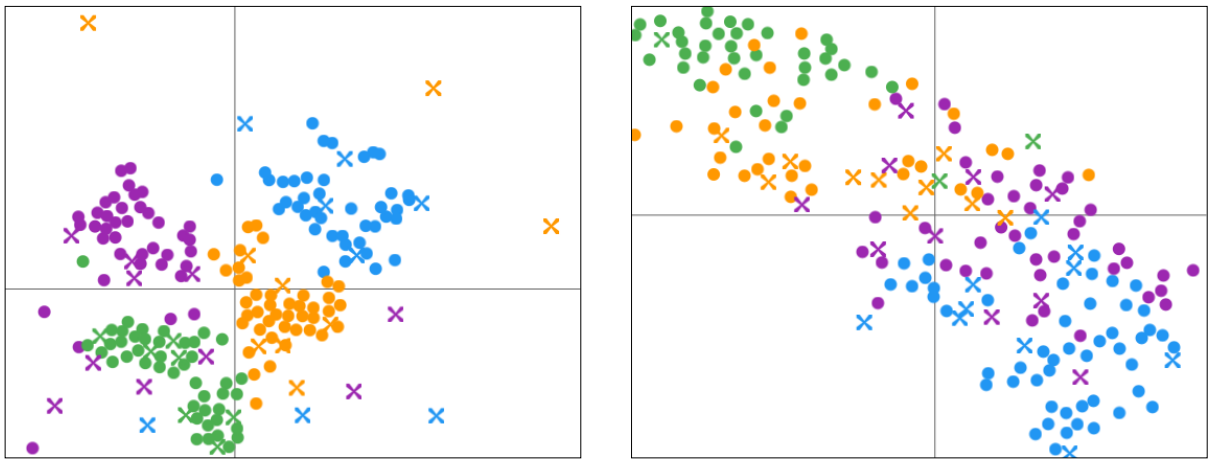
(b) WB.

Figure 5.3: Parallel coordinates visualisations of (a) the MNIST4 and (b) the WB datasets. Instances are colour-coded by class. Instances with confirmed labels are shown as thick lines, suggestions from the classifier are shown as thin lines.

labelling candidates. There, selection strategies were first grouped into *data-centred* and *model-centred* strategies. Data-centred strategies focus on the characteristics of data instances and include Dense Areas First, Centroid First, Equal Spread, Cluster Borders, Outliers, and Ideal Label. Model-centred strategies rely on visual feedback of the current state of the classification model and include Class Distribution Minimisation, Class Borders, Class Intersection, and Class Outliers. In addition to the strategies defined by Bernard, Hutter et al. [2018], in this study, another strategy was observed, which was named Visual Centre. Here, users would select instances in the centre of the visualisation they were currently focussed on.

5.3 Research Questions and Hypothesis

A comparative experiment was conducted to evaluate the effectiveness of three individual visualisation techniques for interactive labelling, based on which records were selected by test users for labelling.



(a) MNIST4. **(b)** WB.

Figure 5.4: Similarity maps of (a) the MNIST4 and (b) the WB datasets. Colours indicate classes. Instances with confirmed labels are shown as crosses, suggestions by the classifier are shown as solid circles.

The three techniques were similarity map, SPLOM with scatterplot for a detailed view, and parallel coordinates. The comparison was both quantitative and qualitative.

The study presented in this chapter, addresses RQ3: *How to compare VA techniques with traditional automated algorithms for building ML models?* This question is further expanded into four research questions.

Research Question 5.1 (RQ5.1): *How do three individual visualisation techniques, (similarity map, SPLOM, and parallel coordinates) compare in terms of accuracy of the resulting classifier?*

Research Question 5.2 (RQ5.2): *How does interactive visual labelling (IVL) with the three visualisation techniques compare to non-interactive labelling based on active learning (AL) selection?*

Research Question 5.3 (RQ5.3): *Which of the three visualisation techniques are rated higher by users in terms of user experience and confidence during selection of records to label?*

Research Question 5.4 (RQ5.4): *Do users adopt different labelling strategy depending on the visualisation being used?*

Furthermore, in this chapter, a hypothesis is formed and tested for these questions.

Hypothesis 3 (H3): *Interactive visual labelling techniques can surpass non-interactive labelling techniques based on active learning in terms of accuracy.*

5.4 Study Design

The user was asked to choose 30 instances for labelling, one instance at a time, each of which was then labelled with its (correct) pre-assigned label from the ground truth.

Regarding RQ5.1, the accuracy of the classifier was computed after each time an instance had been chosen for labelling, using the current training set (i.e. the set of records with confirmed labels at a particular point in time). The accuracy is simply the number of correct predictions divided by the total number of predictions. This experiment was concerned with which instances users chose to label, not

with the actual labels which were then assigned. Hence, users were not actually asked to assign a label, simply to confirm the correct label from the ground truth (see Figure 5.2). To this end, after a user had chosen an instance to label, a pop-up window appeared showing the (pre-assigned) label for that instance, and was simply asked for confirmation. Once the label had been confirmed, the classifier ran in the background to refresh suggested labels for currently labelled instances. Participants were provided neither with guidance nor with any active learning suggestions about which instance to label next, but were asked to choose freely, and without time constraints. Participants were also not informed about the accuracy of the ML model as they worked, but they were shown a chart about accuracy after they had finished working with each dataset.

Regarding RQ5.2, the three active learning algorithms were run for each dataset, the accuracy of the resulting classifier was calculated for each step, and then averaged over all three AL algorithms. This provided the baseline for comparison. The ratings for RQ5.3 were collected after the three visualisation had been labelled for each dataset. The labelling strategies used by each user for RQ5.4 were determined by analysing the thinking aloud protocol, screen recording, and interview responses.

5.4.1 Datasets

Three datasets were used in this study. The first dataset is a two-class subset of the classic MNIST dataset [LeCun et al. 1998], comprising images of hand-written digits in one of two classes: 0s and 1s. It was used to explain mVis to the participants in the tutorials phase of east test session. The 784 dimensions of the original dataset were reduced to 12 by PCA [Jolliffe 2002] and named D1 through D12. The test dataset comprised 200 records with 100 records in each class. This dataset will be referred to as the MNIST2 dataset and is shown in Figure 5.1.

The second dataset is an MNIST dataset with 50 records in each of four classes (200 records total), representing the digits 0, 1, 6, and 7. Like the first dataset, this dataset was reduced to 12 dimensions with PCA. This dataset will be referred to as the MNIST4 dataset. Figure 5.3a and Figure 5.4a show this dataset in parallel coordinates and a similarity map.

The third dataset is a socio-economic statistical dataset published by the World Bank [TWB 2018]. Each record is a country. The ten dimensions represent attributes such as Urban Population, Life Expectancy, and Access to Electricity. The 192 records (countries) are classified (unevenly) into one of four economic classes: *lower income*, *lower-middle income*, *upper-middle income*, and *high income*. This dataset will be referred to as the WB dataset. Figure 5.2, Figure 5.3b, and Figure 5.4b show this dataset in SPLOM with scatterplot, parallel coordinates, and a similarity map.

5.4.2 Participants and Setup

The study was carried out in a quiet lab. Ten participants were initially recruited for the study, but one was later eliminated from the analysis due to technical problems. Of the nine remaining participants, three were female and six were male, with a median age of 29 years. All participants were familiar with machine learning and scatterplot visualisations. Two-thirds (6 of 9) were familiar with SPLOM and parallel coordinates. Two-thirds (a different 6 of 9) had previous experience in labelling multivariate datasets.

During their test session, participants were asked to think aloud, and to ask questions if they experienced any difficulties. At the end of the session, participants were encouraged to make suggestions for improvement. On average, each session lasted around 55 minutes, with the shortest and longest being 43 and 78 minutes, respectively. All sessions were captured by screen recording, and three sessions were additionally recorded with an external video camera for later analysis.

5.4.3 Procedure and Tasks

The test procedure with each participant comprised of four phases:

1. Opening: Introduction and background questionnaire.
2. Tutorial: Demonstration of mVis and practice with the MNIST2 dataset.
3. Test Session: Six experimental conditions, labelling each of the two datasets with each of the three visualisations.
4. Closing: Interview with the participant.

In the first phase, the facilitator explained the purpose of the study and the participants then filled out a background questionnaire. The questionnaire included four binary (yes/no) questions. In these questions, it was asked whether the participant had used machine learning algorithms, scatterplots, SPLOM, and parallel coordinates.

In the second phase, The facilitator first demonstrated the functionality of mVis with the MNIST2 dataset, explaining each of the three visualisation techniques and labelling two of the records. Then, users were asked to label a further 28 records by using all three visualisations.

In the third phase, each test user performed the labelling task for each of the two datasets (MNIST4 and WB) with each of the three visualisations (similarity map, SPLOM with scatterplot, and parallel coordinates). Each visualisation was maximised to full screen. The presentation order of these six experimental conditions was grouped by dataset but otherwise counterbalanced, as can be seen in Table 5.1. In each experimental condition, the test participant was asked to choose 30 instances for labelling (one after the other), which were then assigned their pre-assigned label (class). The experimental conditions were grouped by the dataset. One dataset was loaded, and labelling was completed with the three visualisations, then the second dataset was loaded for the final three visualisations. After each dataset had been explored with all three visualisations, test participants were asked to rate their experience and confidence in labelling the records for each visualisation:

Q_1 From 1 to 5, how do you rate the labelling experience with {visualisation technique}?

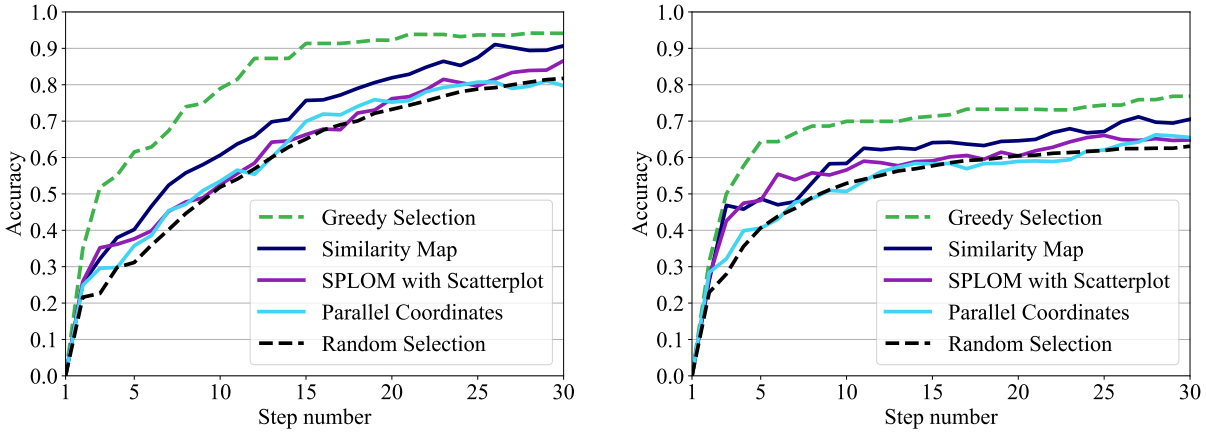
Q_2 From 1 to 5, how confident were you when selecting a new record with {visualisation technique}?

where 1 was the worst and 5 the best rating. In the Q_1, it was clarified to participants to rate the experience of interactive labelling and not the ease of the user interface or other aspects.

Finally, in the fourth phase, the facilitator interviewed the test participants about their experience and encouraged them to offer any feedback or suggestions they might have.

	C_1	C_2	C_3	C_4	C_5	C_6
TP_1	M-S	M-X	M-P	W-S	W-X	W-P
TP_2	M-S	M-P	M-X	W-S	W-P	W-X
TP_3	M-X	M-S	M-P	W-X	W-S	W-P
TP_4	W-P	W-S	W-X	M-P	M-S	M-X
TP_5	W-P	W-X	W-S	M-P	M-X	M-S
TP_6	W-X	W-P	W-S	M-X	M-P	M-S
TP_7	M-X	M-P	M-S	W-X	W-P	W-S
TP_8	W-X	W-S	W-P	M-X	M-S	M-P
TP_9	M-P	M-S	M-X	W-P	W-S	W-X

Table 5.1: The presentation order of experimental conditions. Each row indicates a test participant and columns indicate the order of test conditions. The first letter indicates the dataset (M for MNIST4 and W for WB). The second letter indicates the visualisation (S for similarity map, X for SPLOM and scatterplot, and and P for parallel coordinates).



(a) Accuracy (MNIST4). **(b)** Accuracy (WB).

Figure 5.5: The accuracy of visual labelling depends on the interactive visualisation technique. The y-axis represents the accuracy, the x-axis is the cumulative number of instances already labelled (step number). Greedy selection (green) represents a theoretical upper limit. Random selection (black) represents a practical lower limit.

5.5 Results

The results of the study will be discussed for each of the three visualisation techniques (similarity map, SPLOM with scatterplot, and parallel coordinates) in terms of the four research questions from Section 5.3.

5.5.1 Similarity Map

In terms of accuracy (RQ5.1), the similarity map outperformed SPLOM with scatterplot and parallel coordinates when using both the MNIST4 and WB datasets (see Figure 5.5).

Comparing with active learning (RQ5.2), the similarity map consistently outperforms active learning in both datasets, as can be seen in Figure 5.6.

Regarding the ratings of users (RQ5.3), the similarity map was rated higher than the other two visualisation techniques, both in terms of labelling experience and selection confidence, as can be seen in

Figure 5.7. Indeed, for labelling experience with the MNIST4 dataset, the mean rating of the similarity map was statistically significantly higher than the other two visualisations. All other differences in mean ratings were not statistically significant.

For both rating questions, the similarity map was rated slightly higher for the MNIST4 dataset than the WB dataset. This could be because the clusters in the MNIST4 dataset were more distinct and visible than those in the WB dataset, as shown in Figure 5.4b. This problem persists even when the projection algorithm for the similarity map is changed from t-SNE to PCA or MDS [Kruskal 1964].

When using the similarity map, the strategies used by participants (RQ5.4) were similar to strategies observed during previous studies [Bernard, Hutter et al. 2018]. In the similarity map, users tended to find distinct clusters from the beginning by using a Centroid First strategy. Therefore, the similarity map technique suffers less from the bootstrap problem (Figure 5.5). After identifying distinct clusters, users tried to find outliers and make clear borders. The second main strategy used by participants was Class Intersection, i.e. selecting records which are in the wrong visual section. These records are closer to a different cluster than their own. Based on the observations, identifying suspected incorrectly labelled records in a similarity map was found by the participants to be a rather well-defined task. Note that the accuracy of these labelling strategies depends on the quality of the similarity map, e.g., how faithfully distances in the high-dimensional data space are preserved in the 2d projection space. An interesting variant for a future experiment would be to include measures for projection quality in the similarity map, for which different visualisation techniques exist (see, for example, [Schreck et al. 2010]).

5.5.2 SPLOM with Scatterplot

Regarding the accuracy of the technique (RQ5.1), SPLOM with scatterplot performed slightly worse than similarity map with both datasets, but slightly better than parallel coordinates with the MNIST4 dataset and similar to parallel coordinates with the WB dataset (Figure 5.5). The advantage of SPLOM compared to similarity map and parallel coordinates was that it suffered less from the bootstrap problem.

SPLOM with scatterplot outperformed the active learning techniques (RQ5.2) for both datasets, as can be seen in Figure 5.6.

Regarding the ratings of users (RQ5.3), the SPLOM with scatterplot technique was rated slightly lower than similarity map and slightly higher than parallel coordinates for both rating questions and with both datasets, as shown in Figure 5.7. However, the only statistically significant difference is the lower mean rating for labelling experience for SPLOM with scatterplot compared to similarity map with the MNIST4 dataset. Regardless of the ratings for both datasets being similar, users stated that selecting candidates with the WB dataset was easier, since the dimension names were semantically meaningful and therefore more understandable.

Regarding labelling strategy (RQ5.4) when using the SPLOM with scatterplot technique, users first attempted to find a scatterplot with well-spread records and then used the Centroid First strategy on this scatterplot. Later, some users selected scatterplots with well-separated clusters. Others preferred to select scatterplots which lacked well-separated clusters and attempted to separate them. In order to find outliers, some users tried brushing and linking. Most users tended to select a single scatterplot and continued to use it instead of changing to a different scatterplot. With the MNIST4 dataset, which lacks semantically meaningful dimensions, users selected a scatterplot with a clearer visual pattern, for example linear. Furthermore, users often selected scatterplots located in the centre of the SPLOM and ignored those in the outer reaches.

In general, users selected scatterplots from the SPLOM which: (a) have a specific pattern (for example, linear), (b) have well-separated classes, (c) have overlapping classes, (d) if the dimensions have semantically meaningful labels they select an interesting pair of dimensions based on the context, (e) randomly select scatterplots located in the centre of the SPLOM.

The disadvantage of the SPLOM with scatterplot technique is that it has many false positives. That is, clusters were not always visible and well separated, which confused some users. Moreover, the SPLOM technique was sometimes overwhelming for users.

5.5.3 Parallel Coordinates

Understanding parallel coordinates was hard for the users, mainly due to their lack of experience with this technique. Participants who were familiar with parallel coordinates performed better and were more confident during the experiment. Identifying patterns was difficult, particularly with the MNIST4 dataset. Furthermore, parallel coordinates tended to be more cluttered, and therefore selections became more random over time. Some users were frustrated when they were forced to select points from parallel coordinates. One of the advantages of parallel coordinates was that it well guided the user's visual attention to extremes (peaks and valleys), enabling the users to identify these values easily. Furthermore, when users attempted to make borders for clusters in one single axis, using parallel coordinates was beneficial. On the other hand, one disadvantage of parallel coordinates was its lack of visual feedback, as stated by some users. Moreover, since users often focused on the centre of visualisation, the ordering of the axes was important when using parallel coordinates. Observations showed that if users rearranged the order of the axes, their experience could improve.

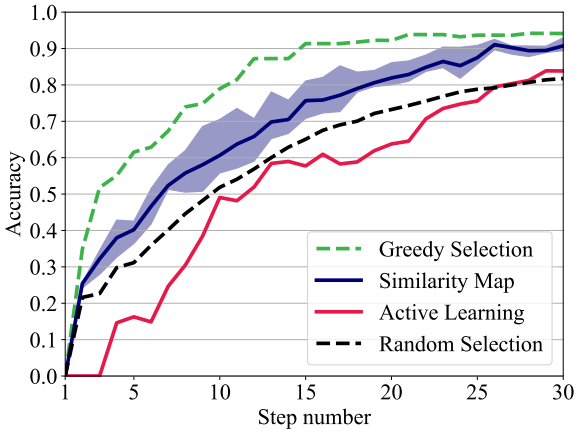
Regarding the accuracy of the classifier (RQ5.1), parallel coordinates performed about as poorly as SPLOM with scatterplot with the MNIST4 dataset and slightly worse than SPLOM with scatterplot with the WB dataset. Parallel coordinates also suffered from the bootstrap problem, due to the users' tendency to select extreme values (peak and valleys) in the beginning and ignore the middle records, which usually included *lower-middle income* and *upper-middle income* countries.

Parallel coordinates outperformed active learning (RQ5.2) in both datasets, although active learning catches up as more instances are labelled (see Figure 5.6).

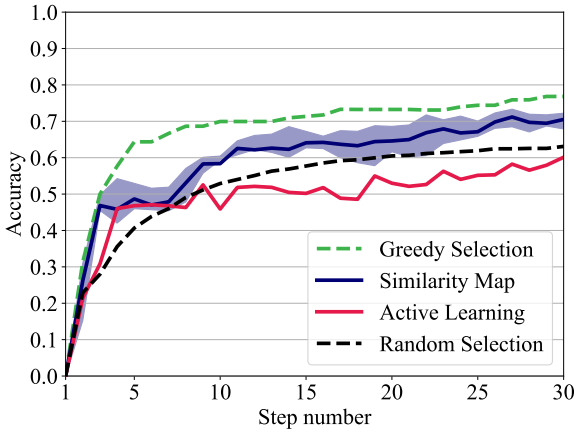
Regarding user ratings (RQ5.3), parallel coordinates received the lowest ratings, both in terms of labelling experience and selection confidence for both datasets, as can be seen in Figure 5.7. The only statistically significant difference is the much lower mean rating for labelling experience for parallel coordinates compared to similarity map with the MNIST4 dataset. However, the mean can be misleading. Half of the users rated parallel coordinates 5 out of 5 when applied to the WB dataset, while the other half rated it poorly. The observations and interviews confirmed that some users strongly preferred parallel coordinates when the clusters were well separated, whilst others favoured other techniques.

In terms of labelling strategy (RQ5.4), participants carried out the following strategies when using parallel coordinates: (a) selected records on a single axis based on their values, (b) focused on a combination of two axes, i.e., a line, (c) focused on the shape of the polyline or general picture in three or more axes, (d) focused on peaks and valleys, (d) randomly selected records on one axis or on a line between two axes. The users' main strategy was to select extreme values in an axis located in the centre of the visualisation. The Density First strategy was a common strategy used by the participants. At the beginning of the tasks, 60 per cent of the participants used the default order of the axes, and 40 per cent customised the order (mVis allows to reorder axes interactively). Users rarely changed the order of the axes afterwards. When using parallel coordinates, users paid less attention to having an equal spread strategy, and therefore, the clusters were more imbalanced. Users also tried to identify class borders, but in the MNIST4 dataset finding such borders was difficult.

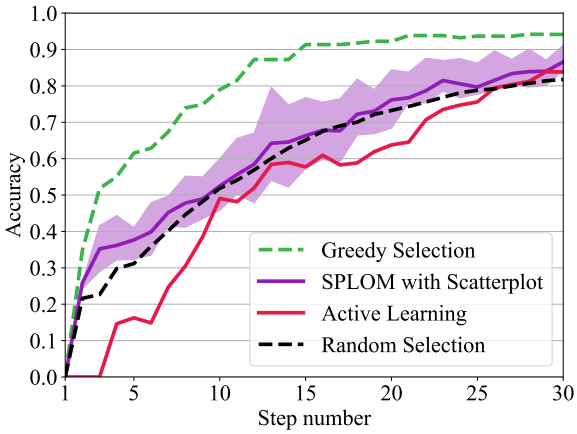
When using the parallel coordinates technique, some users occasionally became frustrated and selected random records located in the visual centre of the plots. A recurring problem was the users' tendency to selecting outliers, leading to the bootstrap problem, as can be seen in Figure 5.5. Furthermore, users selected higher values (peaks) more than lower (valley) values which lead to an imbalance in the selection of peaks and valleys. When using parallel coordinates, users deployed the Ideal Labels strategy more than when using other techniques.



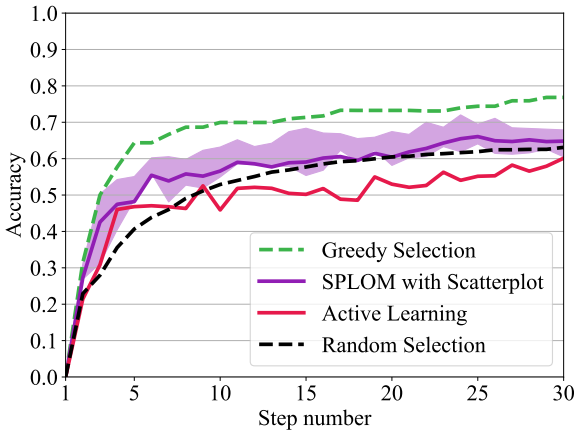
(a) Similarity map (MNIST4).



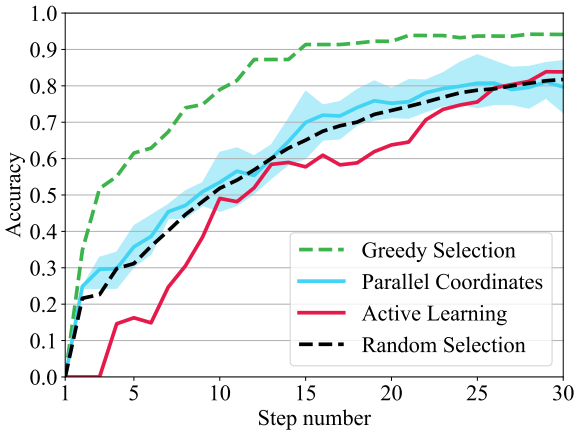
(b) Similarity map (WB).



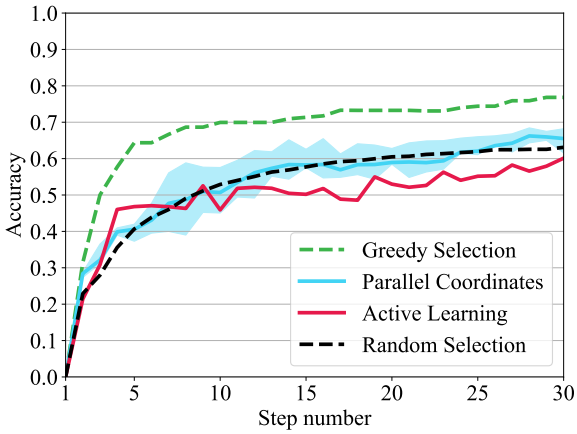
(c) SPLOM with scatterplot (MNIST4).



(d) SPLOM with scatterplot (WB).

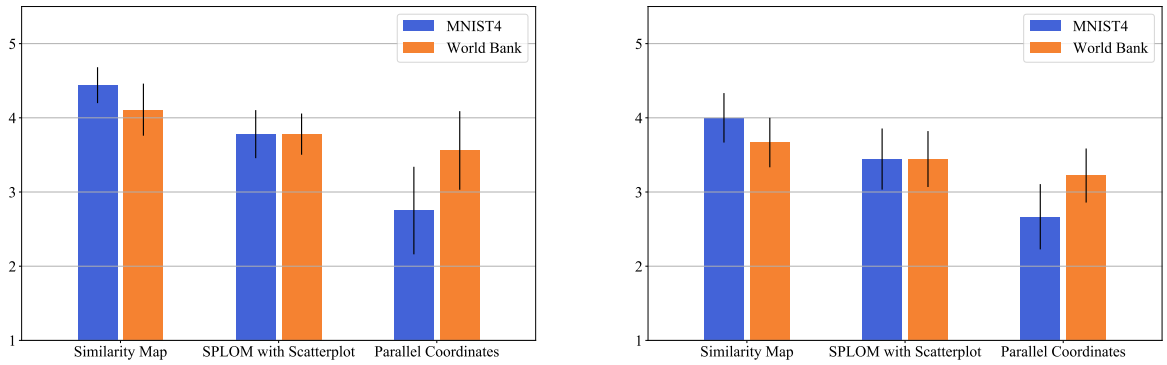


(e) Parallel coordinates (MNIST4).



(f) Parallel coordinates (WB).

Figure 5.6: Accuracy of the three interactive visual labelling techniques compared with active learning (red) for the MNIST4 and WB datasets. The semi-transparent coloured areas show the 25% and 75% quartiles.



(a) Labelling experience.

(b) Selection confidence.

Figure 5.7: Mean ratings given by the test users for (a) labelling experience and (b) selection confidence for each of the three visualisations on a scale of 1 (worst) to 5 (best). Black lines represent standard error.

5.6 Discussion

The results of the study are promising as they show that the classification performance of interactive visual labelling techniques can outperform those of active learning selection strategies. Therefore, this chapter proves that Hypothesis 3 is true. As shown in Figure 5.6, with the WB dataset, all three visualisations perform better after around 10 labelled instances than active learning. In contrast, with the MNIST4 dataset, active learning catches up with the interactive visual labelling techniques as the number of labelled instances increases.

Only a very limited number (9) of test users participated in this study. It would need to be repeated with a much larger number of test users, in order for the results to be more generalisable. The results were also obtained for very specific choices of visualisation and datasets, and their generalisation would require additional validation. Labelling a dataset can be a dull task. Three participants mentioned interactive visual labelling is enjoyable and feels like playing a game.

Some visualisations appear to be better suited to interactive visual labelling than others. The similarity map seems to be the preferred view for labelling. This can be attributed to the fact that the similarity map reduces data, gives an overview of the similarity relationship, and is less complex than SPLOM with scatterplot or parallel coordinates. However, it was observed that when some example labelling is already available, some users prefer to use SPLOM with scatterplot for a more detailed insight into the high-dimensional data space and for label selection. It was also observed that users who are familiar with parallel coordinates perform better and are more confident using it for label decision making.

Parallel coordinates and SPLOM are suitable for finding relationships between dimensions, identifying clusters, and exploring data to make sense of it. Visualisation of the labelled data in parallel coordinates could be improved. As the number of labelled instances increases, it can become overwhelming for the user to find the next instance to label. A problem found in all three visualisation techniques is that of false labelling. When an instance is close to a specific cluster, the user believes the instance belongs to that cluster and does not select it for labelling.

Regarding differences in the two datasets, it was observed that the MNIST4 dataset appeared very cluttered in the parallel coordinates visualisation, and patterns were difficult to discern. Therefore, the results for this test condition may have suffered. In the WB dataset, the dimensions had semantically meaningful names, and users felt more comfortable choosing the axes in the parallel coordinates visualisation and choosing a particular scatterplot from the SPLOM. For example, users often chose the *Access*

to Electricity axis for labelling *low income* countries.

It is interesting to observe in Figure 5.6 that active learning often performs worse than random selection, at least in terms of the simple metric of the ML model accuracy. However, this study only looked at the first 30 labelled instances and AL strategies often start poorly (bootstrap problem), but outperform random selection in later phases [Attenberg and Provost 2011; Kottke et al. 2017].

In terms of improvements, one user mentioned a lack of control over the arrangement of scatterplots within a SPLOM. Another user mentioned that parallel coordinates and SPLOM might be adapted to show the most "important" dimensions. Such an idea is presented in Section 6.5. Active learning was also mentioned by a participant as an additional form of visual guidance [Ceneda et al. 2016] for visualisation techniques. Another participant was curious to see the accuracy of the classifier after the selection of every instance, together with the number of already labelled instances from each class.

Chapter 6

Multimodal Interaction for Data Analysis

"The best and most beautiful things in the world cannot be seen or even touched - they must be felt with the heart."

[Helen Keller, American author, 1880-1968]

Interaction is the heart of visual analytics. VA with traditional interaction devices, such as mouse and keyboard, can help the analyst to explore and build ML models. But how other interaction modalities, including multi-touch devices and eye-tracking can foster this process? This chapter presents two novel techniques for using multi-touch interfaces for exploring regression models collaboratively. Moreover, it is proved that indirect gaze input can be an additional interaction method for exploring multivariate datasets.

6.1 Introduction

Interaction an essential part of both *Information Visualisation (InfoVis)* and *Visual Analytics (VA)*. In one of the early definitions for InfoVis, Card et al. [Card et al. 1999] describe visualisation as the "mapping of data to a visual form that supports human interaction in a workplace for visual sense-making". Although the concept of interaction in InfoVis and VA has a long history [B. Lee et al. 2012], novel device and display technologies, and novel multimodal interaction possibilities [B. Lee et al. 2018] including gesture recognition, eye tracking, or data physicalisation offer new possibilities.

Visualisation techniques should be adapted according to the type of data, user task, and display medium. For example, scatterplots allow an analyst to quickly recognise patterns and relationships between any two of n dimensions of a multivariate dataset and have become a common technique for data visualisation of multivariate datasets. It is also possible to plot all the possible bivariate projections of a dataset, resulting in a matrix of n^2 scatterplots, called a *scatterplot matrix (SPLOM)* [Cleveland 1993]. In a related technique, drawing all n dimensions as vertical axes next to each other and drawing each record as a polyline intersecting each axis, produces a chart known as a *parallel coordinates* plot [Inselberg 1985]. This chapter presents three different types of interaction modalities that can facilitate data exploration, labelling, and analysis.

Section 6.3 presents challenges and solutions for collaborative and single-task multi-touch interaction on large vertically-mounted high-resolution displays. The techniques presented are well-suited for collaborative analysis tasks with scatterplots and SPLOM. They are potentially generalisable for other data exploration and visual analytics practices but require further implementation and evaluation.

There are various direct and indirect interaction techniques to explore multivariate datasets on a large display. In Section 6.4, an affordable technique using a secondary wireless handheld device is introduced.

By using this technique, the analyst can perform traditional visual analytics tasks including selection, brushing, and linking on a handheld device which is projected on the large display. As a proof of concept, the technique is implemented for exploring scatterplots in a multiple linked-view application.

Finally, Section 6.5 presents an eye-tracking based solution for visual exploration of parallel coordinates. As a proof of concept, a framework and heuristics for interactive axis reordering is introduced.

6.2 Research Questions and Hypothesis

This chapter addresses RQ4: *How to use non-traditional interactions to improving building and exploring the ML model and foster collaboration in teams?* Various devices can be used to interact with visual analytics techniques. In this thesis, the focus is on large multi-touch displays, and eye-trackers. Therefore; RQ4 is broken down to two research questions.

Research Question 6.1 (RQ6.1): *How to use large multi-touch displays to explore the machine learning model and foster collaboration in teams?*

Research Question 6.2 (RQ6.2): *How to use indirect gaze input to explore a multivariate dataset?*

According to these two questions, two hypotheses are formed and tested in this chapter.

Hypothesis 4 (H4): *Large multi-touch displays facilitate collaborative analysis of ML models.*

Hypothesis 5 (H5): *Indirect feedback from gaze can improve interaction and visual exploration of a multivariate dataset.*

6.3 Large Vertically-Mounted Multi-Touch Displays

Large high-resolution displays are becoming an affordable option for the visualisation of data [Reda et al. 2015]. Large displays have proved to be effective for tasks such as comparative genomics analysis [Ruddle et al. 2013], graph topology exploration [Prouzeau et al. 2016a], and sensemaking [C. Andrews et al. 2010]. Large vertically-mounted (landscape-orientation) high-resolution multi-touch displays are particularly effective for collaborative analysis by small teams. However, previous research has often focused on horizontally-mounted tabletop surfaces or vertically-mounted displays with more distant interaction [Jakobsen and Hornbæk 2014]. In this chapter, a set of user interactions to support SPLOM analysis on vertically-mounted displays are introduced. These techniques help analysts to efficiently select a scatterplot from SPLOM and explore it collaboratively.

Some physical and virtual interactions with large displays were described in the previous literature. Modalities range from natural interactions like speech, body tracking, gaze, and gestures to the use of secondary control devices like mobile phones, tablets, or Wii controllers [Khan 2011]. Of these, multi-touch interactions provide a fluid and intuitive interface suitable for up-close interaction in front of the display by small groups. Although there are studies about collaborative interaction with large displays (e.g. [Vogt et al. 2011; P. Isenberg et al. 2009]), they usually focus on single-user interaction [Liu et al. 2017]. Since typical multi-touch interactions do not support collaboration, more research needs to be done on cooperative gestures, modalities and the dynamics of group work around these devices. Cooperative gestures are known to enhance the sense of teamwork and increase the participation of team members [Morris et al. 2006].

Screen size and resolution are particularly important for information visualisation of multivariate datasets. Having a large display allows multiple, linked views, such as SPLOM and parallel coordinates [Inselberg 1985] to be provided simultaneously. If the screen is not high-resolution, the user experience of near distance interaction decreases significantly. For instance, on screens with less than sixty pixels

Figure 6.1: Two users collaboratively analyse a dataset on a large vertically-mounted multi-touch screen. User A on the left drags a Regression Lens, while user B on the right adapts the degree of the regression model using the floating toolbox. The display is an Eyevis 84-inch 4K/Ultra-HD 60Hz multi-touch LCD monitor with a resolution of 3840×2160.

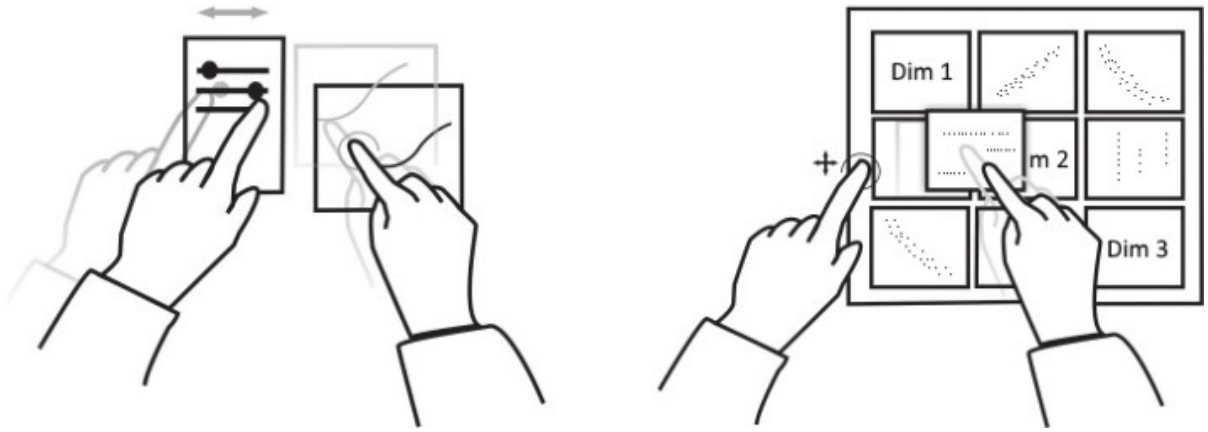
per inch, the user is not able to read from the screen up-close [Ashdown et al. 2010]. Furthermore, users can make more observations with less effort using physical navigation (e.g., walking) rather than virtual [Reda et al. 2015]. More screen space can be used to either provide a better overview of a dataset or to provide more details of a portion of it. For example, users can see both an entire SPLOM, specific scatterplots, and parallel coordinates plots at the same time. As a result, users may have the opportunity to gain more insight into large datasets.

Previous studies [Jakobsen and Hornbæk 2014] suggest that vertically-mounted displays are more suited to parallel tasks within a group, due to reduced visual distraction and the possibility to share information through physical navigation like turning the head or walking. On tabletop displays, if users are not on the same side of the table, the shared view often needs to be reoriented.

This section addresses the design gap between standard interaction techniques for large, multi-touch displays and advanced interaction techniques and visual feedback for collaborative scatterplot and SPLOM analysis. Design concepts for such interaction techniques have been implemented as a proof of concept and are presented. The techniques include scatterplot selection from SPLOM, collaborative regression model analysis, and an extension of the Regression Lens [Shao, Mahajan et al. 2017] to include a floating toolbox. As a proof of concept, the techniques are developed on a large display.

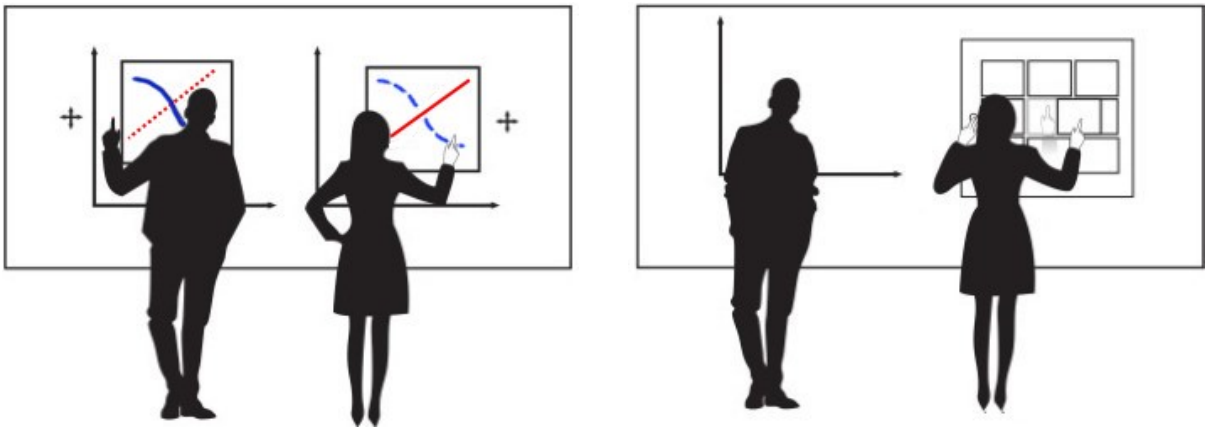
6.3.1 Proposed Interaction Techniques

Current standard multi-touch interaction techniques are not designed for collaboration on vertically-mounted high-resolution displays [Liu et al. 2017]. Here, by proposing single-user and collaborative interactions for the analysis of scatterplots and SPLOM on such devices, Hypothesis 4 is tested. Some of the interaction techniques are based on the concept of the Regression Lens [Shao, Mahajan et al.



(a) Interacting with a Regression Lens. **(b)** Selecting a new scatterplot.

Figure 6.2: A user is drags a Regression Lens with the right hand while adjusting the lens with the left hand (a). A user drags a scatterplot with the right hand while panning through the SPLOM with the left hand (b).



(a) Analysing scatterplots collaboratively. **(b)** Passing interesting scatterplots.

Figure 6.3: Two users collaboratively analyse a scatterplot (a). Both users create a regression model for a subset of selected data. The created regression models are displayed in their partner's respective lens as well, supporting comparison of local regressions. In (b), one user analyses a scatterplot, while their partner selects interesting plots in the SPLOM and passes them over by holding the background and swiping the right hand.

2017], which supports real-time regression analysis of subsets of a scatterplot through lens selection and manipulation. With Regression Lens, a user can select a local area in a scatterplot and observe the regression model of selected points [Shao, Mahajan et al. 2017]. Shao, Mahajan et al. [2017]. proposed operations to adjust and manipulate the regression model shown in the Regression Lens, such as changing the degree of the regression model or inverting its axes. Figure 6.1 illustrates some of the suggested collaborative gestures on an 84-inch 4K/ULTRA-HD@60HZ multi-touch LCD monitor produced by Eyevis [eyevis 2018]. The user on the left finds interesting scatterplots and passes them to the user on the right. The user on the right analyses the plots using the Regression Lens [Shao, Mahajan et al. 2017]. In the rest of this section, four interaction designs for both collaborative and single scatterplot analysis are introduced. Later in Section 6.3.2, an implementation of these techniques is demonstrated.

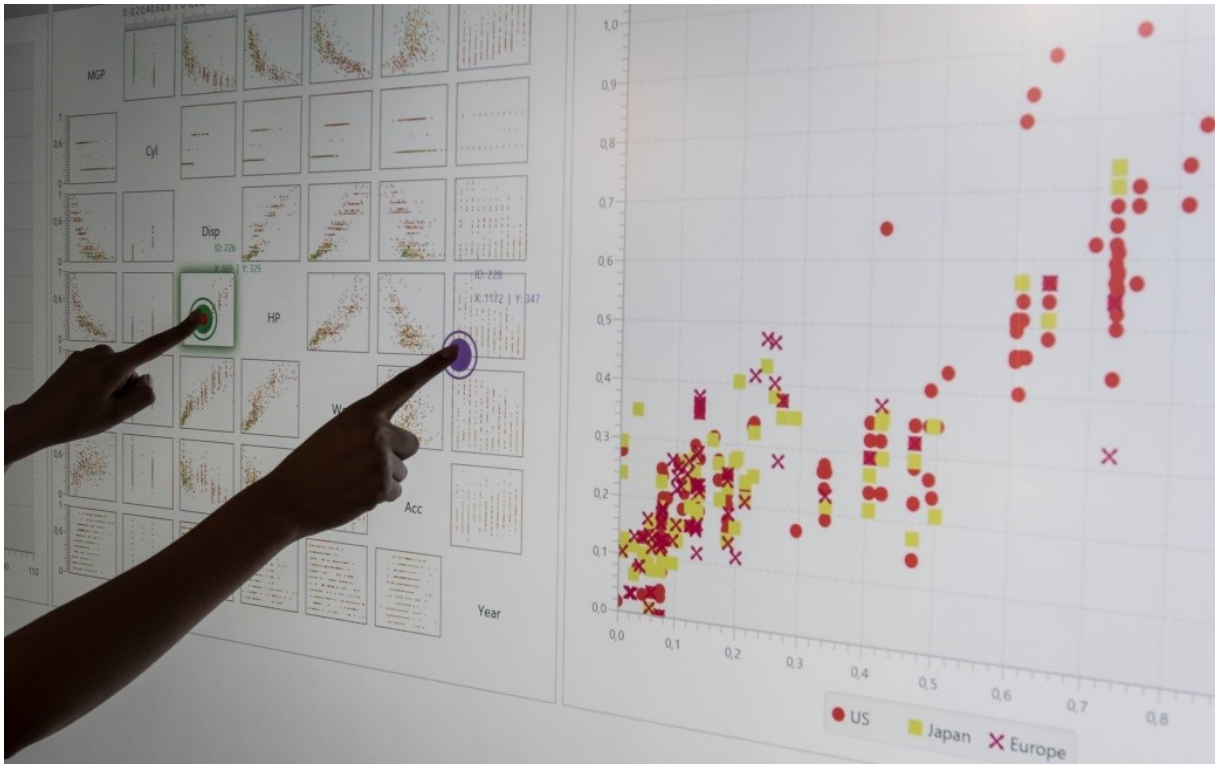


Figure 6.4: A user selects a scatterplot of interest from SPLOM by touching and holding the left hand on the scatterplot. Swiping with the right hand then passes the selected scatterplot to the right hand side of the display for more detailed analysis.

6.3.1.1 Lens and Floating Toolbox

Magic lens techniques like DragMagics [Prouzeau et al. 2016b] and BodyLens [Kister et al. 2015] are used to explore local regions in a visualisation. An extended version of the basic lens concept provides for more fluid interaction with large multi-touch displays. For instance, as shown in Figure 6.2a, after a region of interest has been selected in a scatterplot using the dominant hand (here the right hand), a toolbox appears next to the other side of the lens (near the non-dominant hand), where the user can use sliders and touch buttons to adjust the lens. For example, the user can change the degree of the regression model. The lens can be dragged with one hand, while being adjusted with the second hand, thus potentially speeding up performance.

6.3.1.2 Two-Handed Interaction with SPLOM

A SPLOM consists of pairwise scatterplots arranged in a matrix, with dimensions typically labelled in the diagonal cells. Since the number of dimensions is usually high, panning and zooming within the SPLOM is almost inevitable. With common multi-touch interactions, the scatterplot or dimension label is dragged to the corner of the SPLOM for panning. It is not feasible to zoom into or out of a SPLOM while dragging another object. Based on two-handed interaction on tablets [Yee 2004], a two-handed technique is proposed whereby the dominant hand is responsible for dragging items, while the non-dominant hand performs common operations. As shown in Figure 6.2b, the user drags a scatterplot around to reorder the plots in the SPLOM. Panning is performed by the non-dominant hand. With this two-handed technique, the interactions needed to reorder scatterplots in SPLOM can be reduced.

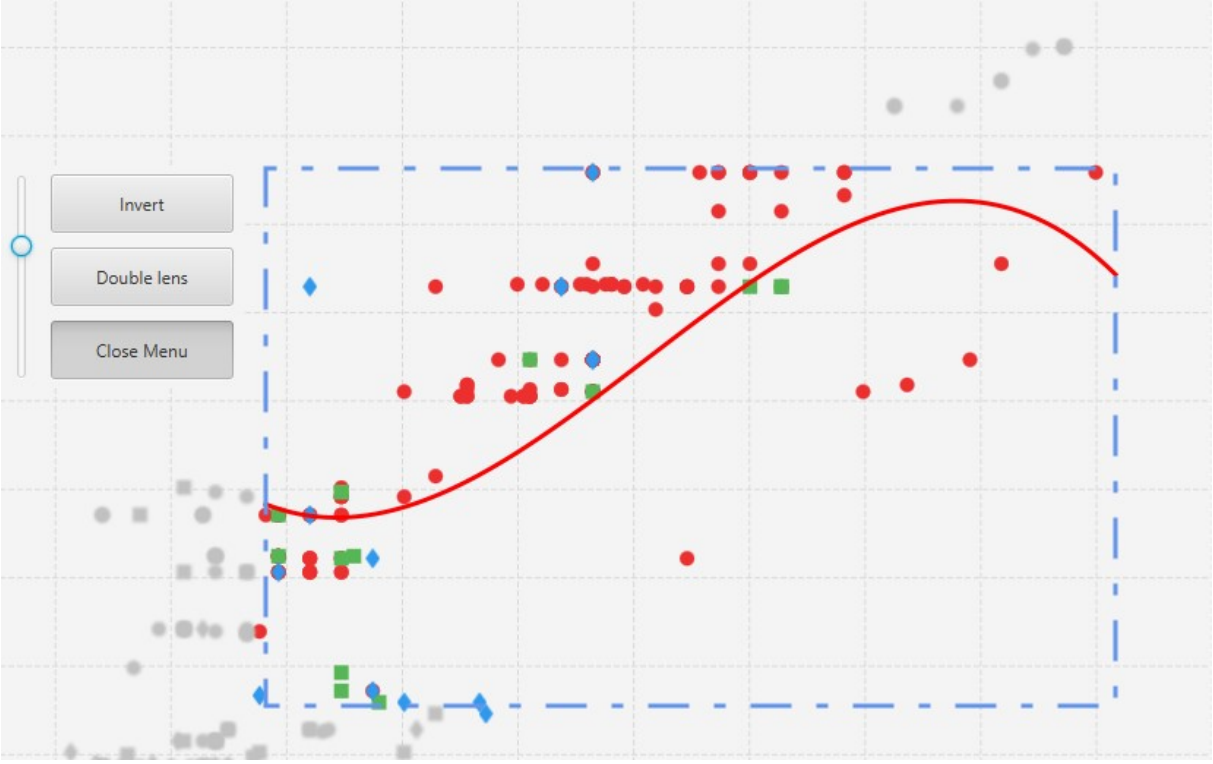


Figure 6.5: A Regression Lens containing a cubic regression model is shown. At the left side of the Regression Lens, a floating toolbox with various options is visible.

6.3.1.3 Collaboration using Gestures

On large vertically-mounted collaborative displays, it is not always desirable to move from one side of the screen to the other to perform a task. Instead, collaborative gestures can be used to pass objects. Based on the ideas of Liu et al. [2017], collaborative gestures on scatterplots are proposed. In Figure 6.3b, the user on the left analyses a scatterplot, while the user on the right selects another scatterplot of interest. By holding the background of the SPLOM with one hand, and swiping with the other hand, the scatterplot is passed over to the partner. The partner can then decide whether or not to load the scatterplot for comparison. This technique can also be used for other tasks. For example, in Figure 6.4, the user selects a scatterplot of interest from a SPLOM by touching and holding it with one hand (here, the left hand) and swipes the other hand in the direction of the analysis panel to load that scatterplot for more detailed analysis.

6.3.1.4 Collaborative Lens

In collaborative analysis, visual feedback plays an essential role. When two analysts work on a vertically-mounted display without proper visual feedback, they need to communicate more and turn their heads more often. A collaborative lens can help ameliorate this issue. As illustrated on Figure 6.3a, the user on the left side of the screen creates a regression lens and regression model in blue. Meanwhile, the user on the right side of the screen creates their regression lens and regression model in red. Both users can see the other user's regression model reflected in their own regression lens.

6.3.2 Implementation

Proof-of-concept interaction techniques for single-user and collaborative analysis of scatterplots and SPLOM have been implemented on a vertically-mounted Eyevis 84-inch multi-touch display with a

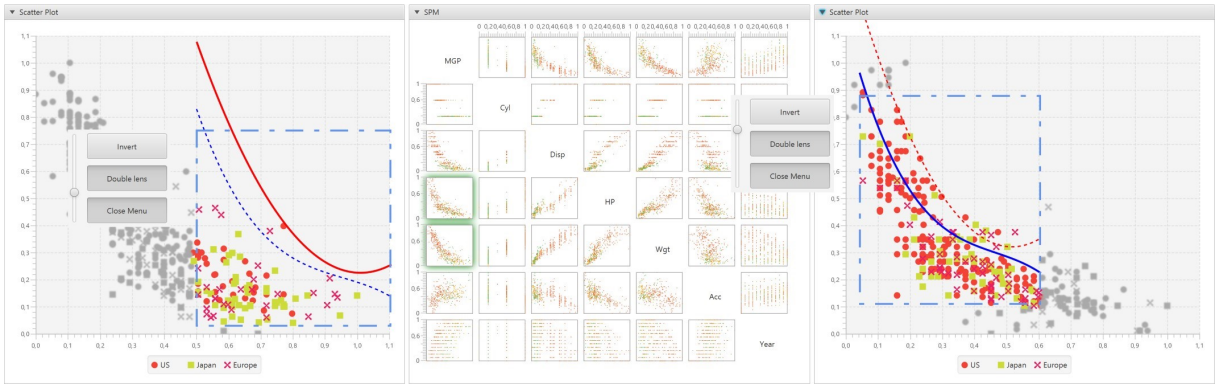


Figure 6.6: The left and right panels are scatterplots for User A (left) and B (right) respectively. The central area of the screen contains a shared SPLOM. User A on the left draws an arbitrary rectangle and is interested in the quadratic regression model of the selected records, shown in red. User B on the right chooses to observe the cubic regression model of the selected area, shown in blue. User A can see the cubic regression model of the right panel in dashed blue and user B can see the left panel regression model in dashed red. Selected scatterplots are highlighted in green in the SPLOM.

resolution of 3840×2160 pixels and a frame rate of 60 Hz. Figure 6.1 demonstrates a typical setup of the implemented application with two users working on the screen.

The prototype application is written in Java, using JavaFX for the user interface and the TUIO [Kaltenbrunner et al. 2005] and the TUIOFX library [Fetter and Bimamisa 2015] for multi-touch interaction. To enable multiple users to work on the same screen with different widgets and user interface elements at the same time, a concept called focusArea from the TUIOFX library is used [Fetter et al. 2017]. The application follows the widely-used Model-View-Controller (MVC) architecture.

6.3.3 Use Case

The use case for the prototype application is to improve interaction with the Regression Lens on multi-touch screens. The developed interaction techniques were tested with the well-known car dataset from the UCI Machine Learning Repository [Lichman 2013].

For the interaction technique shown in Figure 6.1, user A (on the left) and user B (on the right) select two different plots from the shared central area containing the SPLOM. For this technique, the user holds and touches a scatterplot with one hand and swipes to the right or left with the other hand to maximise it. This technique is elaborated in detail in Section 6.3.1.3. After that, users A and B select an area in the scatterplot separately and toggle the Collaborative Lens option in the Floating Toolbox. As described in Section 6.3.1.4, each user is now able to observe the regression model of the other user in their regression lens. Figure 6.1 shows two users working side by side on a large vertically-mounted multi-touch display, after creating two separate Regression Lenses and toggling to the Double Lens option. The exact state of the screen is shown in Figure 6.6. A single Regression Lens with a floating toolbox is visible in Figure 6.5.

6.3.4 Large Multi-Touch Displays Findings

The concepts described in this section are first designs of appropriate touch interaction for the visual interactive analysis of scatterplot data on large vertically-mounted high-resolution multi-touch displays. The interactions support small-group collaborative analysis, by exchanging patterns or settings from one user's view to the others. Therefore; Hypothesis 4 is successfully tested. The interaction design is currently based on user selections, but is generalisable to other basic techniques. The interaction

Figure 6.7: The analyst in the middle is explaining the dataset on the large screen to other team members in a meeting. She is using a secondary handheld device to perform selection on an interesting part of the data.

techniques have been implemented as a proof of concept. They still need to be evaluated with real users and real tasks as part of future work. Mapping out the design space for this combination of visualisation and display device may well yield further interesting interaction designs.

6.4 Secondary Handheld Device

Since both SPLOM and parallel coordinates charts aim to visualise the whole dataset in one view, a large number of pixels are needed on the screen, which motivates the use of larger displays. Such displays are commonly used in meeting rooms for decision making and presentation. Techniques developed by researchers to interact with large displays in VA applications include natural language [Srinivasan and Stasko 2018], multi-touch, full body [Kister et al. 2017], and secondary handheld devices [Tsandilas et al. 2015].

Each of these interaction modalities has strengths and weaknesses. Natural language is a powerful tool to interact with the screen from afar, but some tasks including data selection cannot easily be performed by this interaction alone. Multi-touch is another popular input modality, but may cause fatigue in a long meeting, since it requires the analyst to interact with the screen up-close. In such situations, using a handheld device connected wirelessly to the display can be a suitable option, visualising the data on the large display while at the same time giving analyst(s) the ability to control views and issue queries from a distance as well as up-close.

This section describes how a multiple linked-view information visualisation application can benefit from adding a secondary handheld device as an additional controller for data exploration. In particular, it is shown how common VA techniques like brushing, linking, and querying can be facilitated using a secondary handheld device. This concept can be generalised to other multiple linked-view applications. Fig. 6.7 shows the implemented system being used in a meeting room.

(a) Scatterplot view on the secondary handheld display.

(b) Parallel coordinates view on the secondary handheld display.

Figure 6.8: Any view can be shown on the secondary handheld device to perform tasks such as brushing and linking from afar.

6.4.1 Proposed Interaction Techniques

It is becoming common for large displays to have wireless connectivity. Even without this feature, it is often possible to attach a wireless adapter to a monitor and then connect it to a secondary handheld device. In this chapter, a 12.3-inch tablet (Microsoft Surface Pro, 5th Gen) is used as a secondary handheld device to interact with a large 82-inch 4K display with wireless connectivity (Samsung 88th series).

The mVis tool presented in Chapter 4 is used for this prototype. The system is dock-based, making it possible to detach any view and drag it to a secondary device. Moreover, it is possible to have two identical views – one on the large display and the other on the secondary handheld device – so that everyone in the room can observe how interactions with the handheld view are performed.

6.4.1.1 Brushing and Linking from Secondary Device

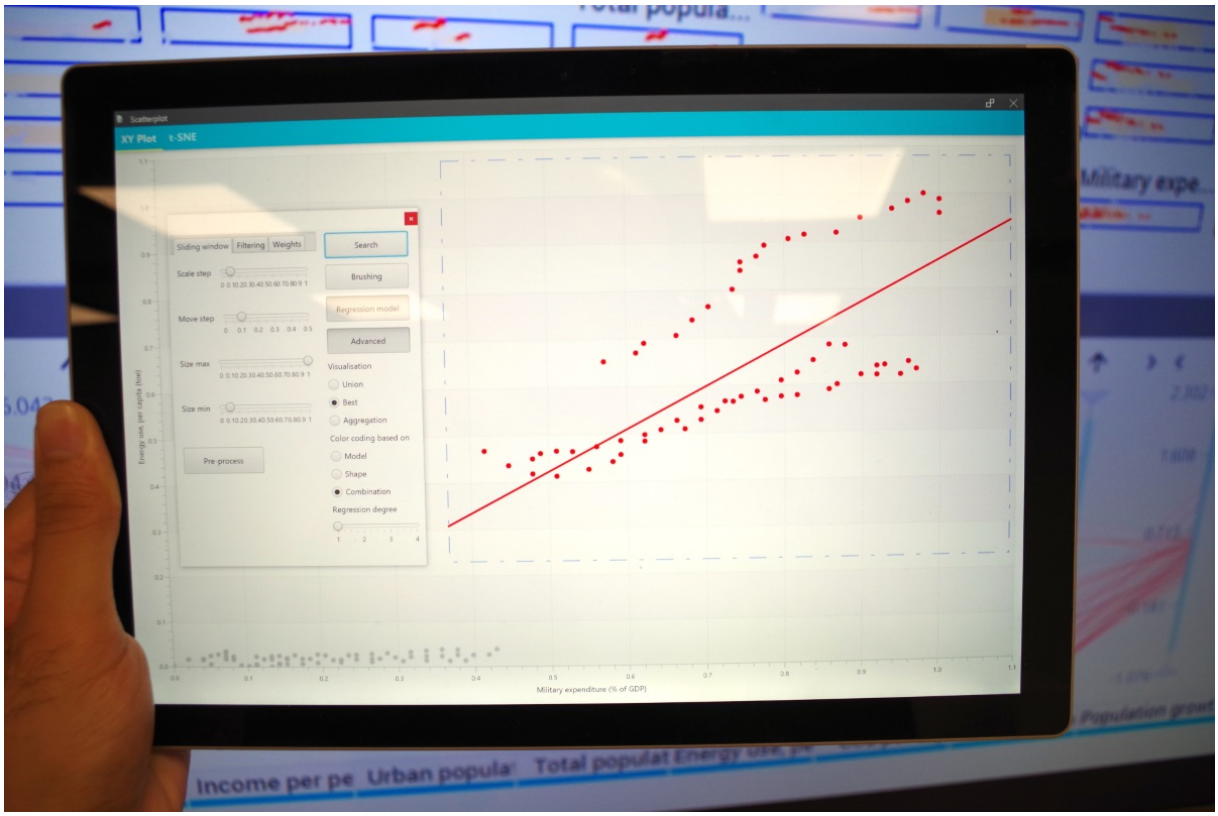
Brushing and linking are standard techniques for interactive visual analysis of large multivariate datasets [Buja et al. 1991]. They refer to the practice of selections (brushing) and changes (linking) in one view being simultaneously reflected in all other views. Collaborative brushing and linking ensures that interactions on the dataset by one collaborator are visible for everyone working on the data [P. Isenberg and Fisher 2009]. Therefore, in the presented system, if a user selects an area on a secondary handheld device using either a scatterplot or parallel coordinates view, the selected records will also be highlighted in the large display. For the other collaborators to have a clear understanding of brushing and linking procedure, the view shown on the secondary handheld device is always projected onto the main screen. This projected view includes both visualisations of the data and interactions performed on each handheld device. Fig. 6.8 demonstrates brushing and linking in scatterplot and parallel coordinates views.

6.4.1.2 Querying from Secondary Device

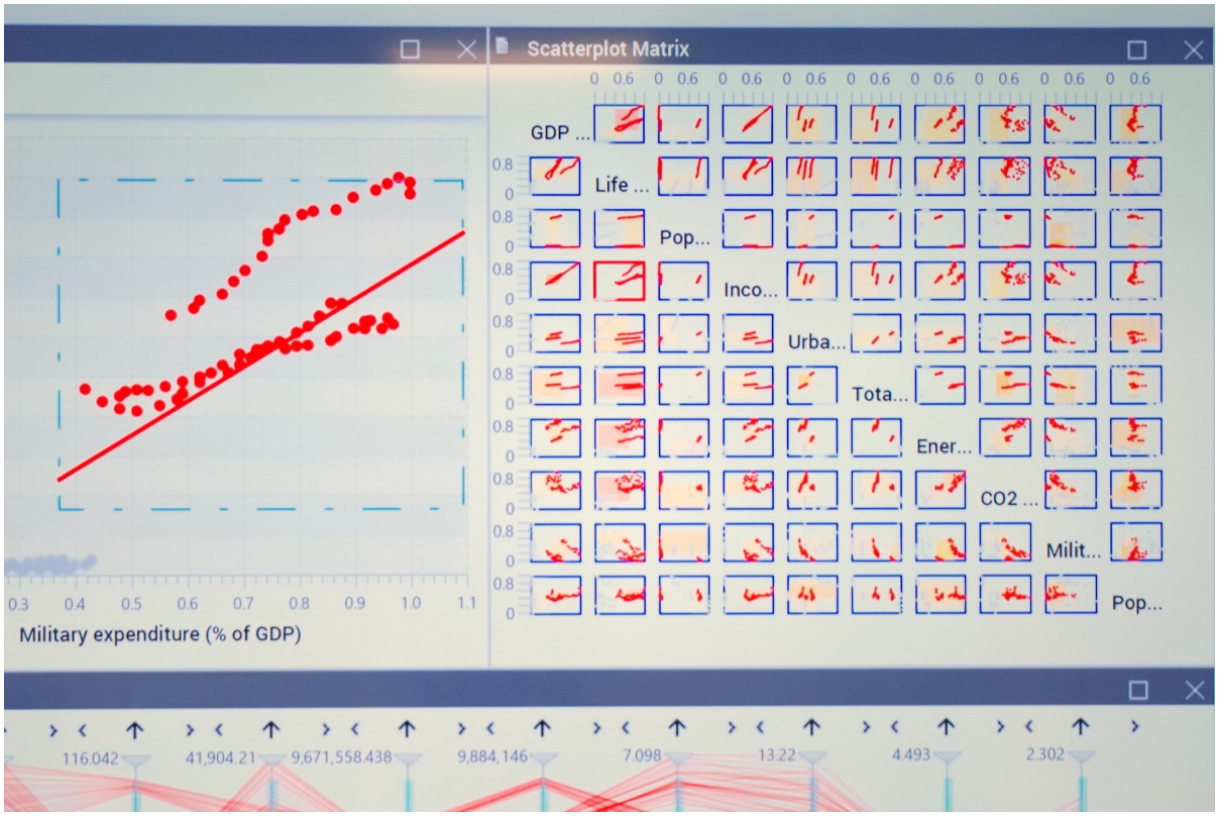
Another common technique in interactive visual analysis is to make a query after selecting an area in a view containing a set of records. In the system, the analyst can select an area in a scatterplot by lasso selection and then perform a search query to find similar areas in other scatterplots in the dataset. It is essential to visualise any such selections on the large display, so all collaborators in the meeting are informed about it. Therefore, an identical view on the large display shows the selection and parameters of the query issued on the secondary handheld device. Fig. 6.9 illustrates how a query on the secondary handheld device is visualised on the large display.

6.4.2 Second Handheld Device Findings

Preliminary experiments show that using a handheld device can be an appropriate proxy to interact with data visualisations on a large display, especially if several users are gathered around it. A secondary handheld device can be easily passed around for individual interaction, while the main display remains in sight. This is particularly useful for more detailed interactions, such as the specification of a query or the exact positioning of a regression lens widget [Shao, Mahajan et al. 2017]. A lightweight solution is possible using component-based application design in conjunction with the multiple-view desktop extension capabilities of current operating systems. Fig. 6.10 shows situations that can benefit from this type of interaction. A natural extension is to provide not one, but multiple secondary handheld devices for distributed, collaborative interaction by team members. To this end, a client-server implementation should be adopted.

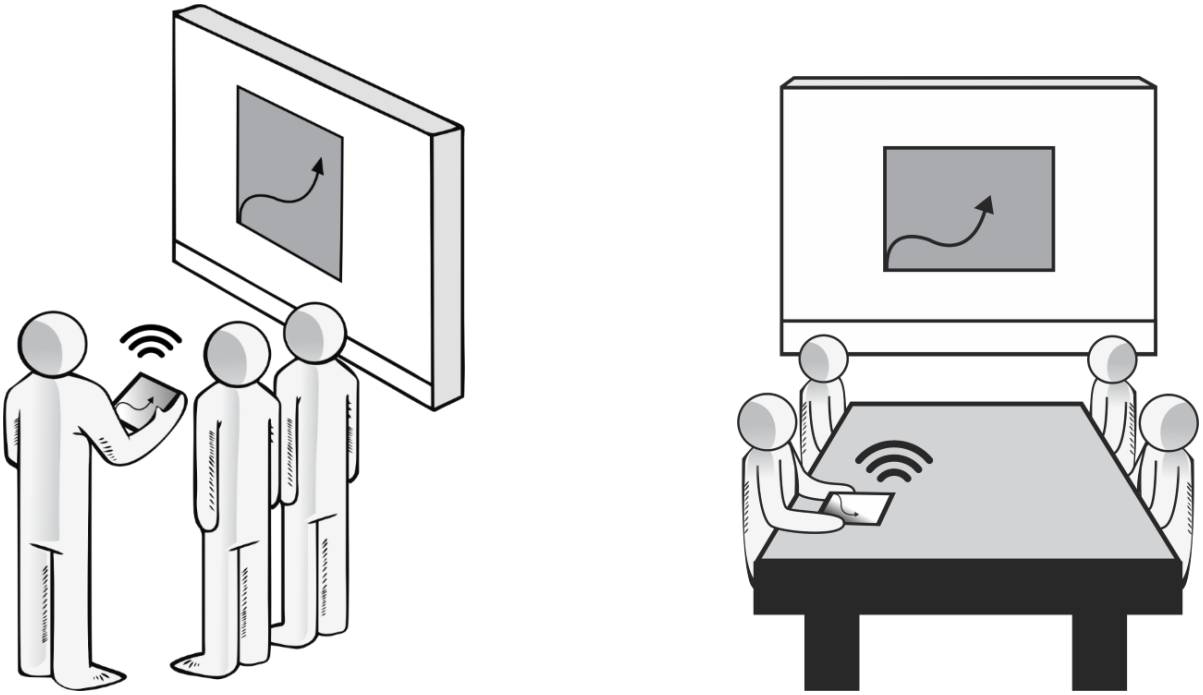


(a) A query performed on the secondary handheld device.



(b) The query and its result on the large display.

Figure 6.9: The user can initiate a query on the secondary handheld device and visualise the results on the large display.



(a) Meeting in front of the screen.

(b) Meeting around a table.

Figure 6.10: Two scenarios in which an analyst can pass over the secondary handheld device to other collaborators.

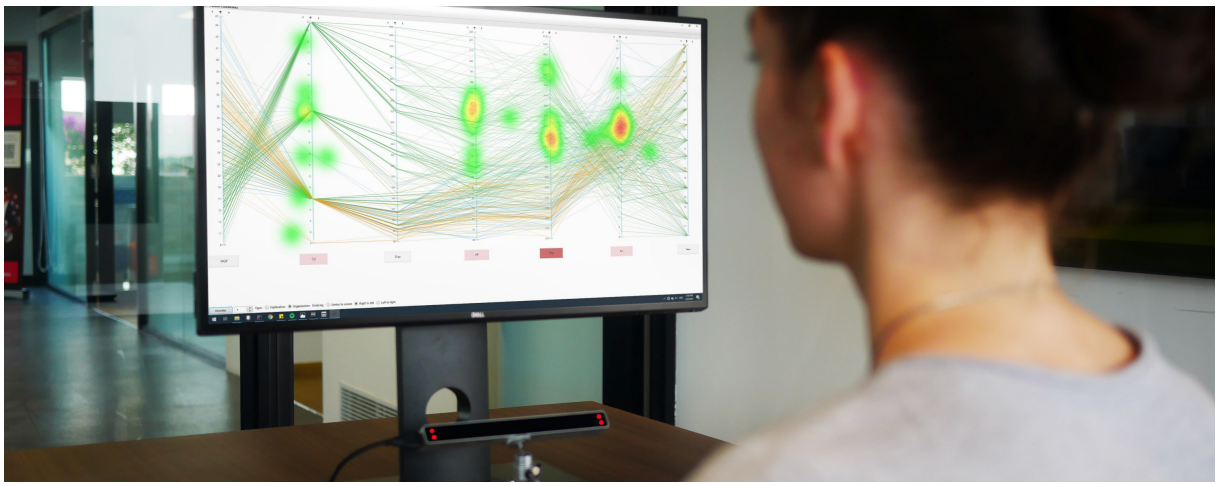


Figure 6.11: An analyst exploring a parallel coordinates plot using an inexpensive eye-tracker.

6.5 Eye-Tracking and Gaze

Parallel coordinates are suitable for exploring correlations between neighbouring dimensions and obtaining a concise overview of an entire dataset (see Chapter 4). Unfortunately, as the number of dimensions increases, the parallel coordinate plot can suffer from visual clutter. In this case, the analyst may overlook an axis or become frustrated by the visual noise of less important ones. Since different users are interested in different parts of the plot, it is essential to provide them with an adaptive ordering method. One way to facilitate ordering is to use an eye-tracker to measure interesting or unexplored parts of the plot. Shao, Silva et al. [2017] showed that using eye-tracking can help an analyst find more patterns in a multivariate dataset in less time. To test Hypothesis 5, two approaches to help the analyst reorder the axes based on an *area-of-interest* (AOI) are presented. Figure 6.11 illustrates the application setup.

6.5.1 Gaze Technique

The mVis system presented in Chapter 4 implements adaptive parallel coordinates which support multiple interaction techniques. The user can invert an axis and filter records based on the values of a dimension by interacting with the axis. For reordering, based on guidance types in visual analytics systems [Ceneda et al. 2016], three main strategies for an adaptive parallel coordinate plot are presented. The basic strategy is to let the analyst manually reorder the plot. Alternatively, by using eye-tracking, the system can guide the analyst more effectively. The system visualises the axes which the analyst explored more than others. Later, the user can either manually reorder the axes based on the provided information, or ask the system to do it automatically.

6.5.1.1 Manual Reordering

In the presented approach, the analyst can perform manual reordering in three ways. Firstly, left and right arrow buttons are provided at the top of each axis. If the user presses the right arrow button, the axis will move one to the right, and all the other axes will shift one to the right. A similar event will occur if the user presses the left arrow button. Secondly, the user can drag and drop an axis onto another axis. As a result, these axes will switch their places. Thirdly, the user can drag an axis and drop it between two other axes.

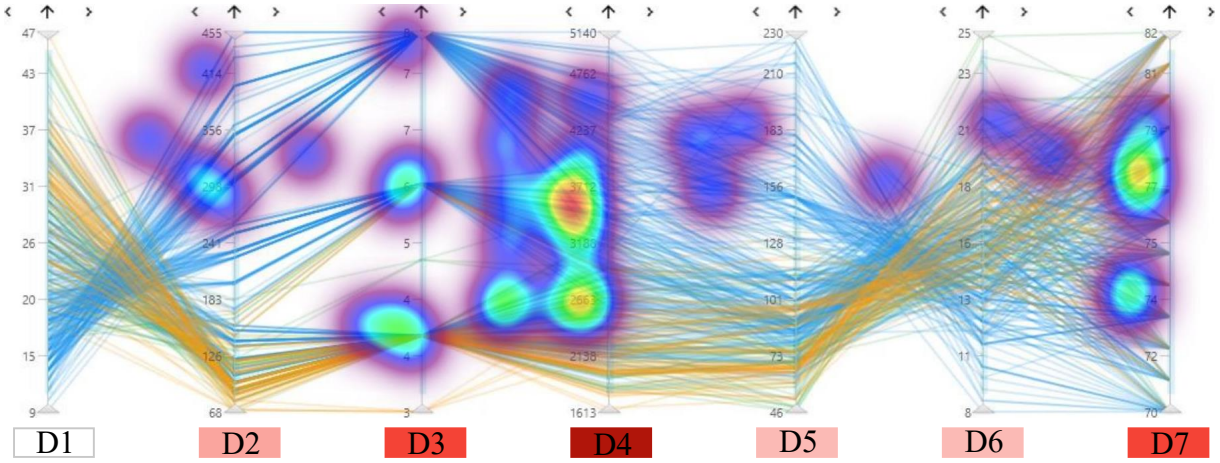
6.5.1.2 Visual Guidance

By using an AOI based approach, the system stores the areas that the user has looked at. There are two types of AOI in the design, (1) the area between two axes, and (2) the axis itself. A heatmap will assist the user in recognising areas that are more explored. While exploring the plot, a heatmap is overlaid on top of the area, either as a transparent colour between two axes or by changing the background colour of the label of the axis. Figure 6.12a shows the (usually hidden) internal heatmap of areas that the user has focused on so far. Figure 6.12b shows the plot after axis reordering.

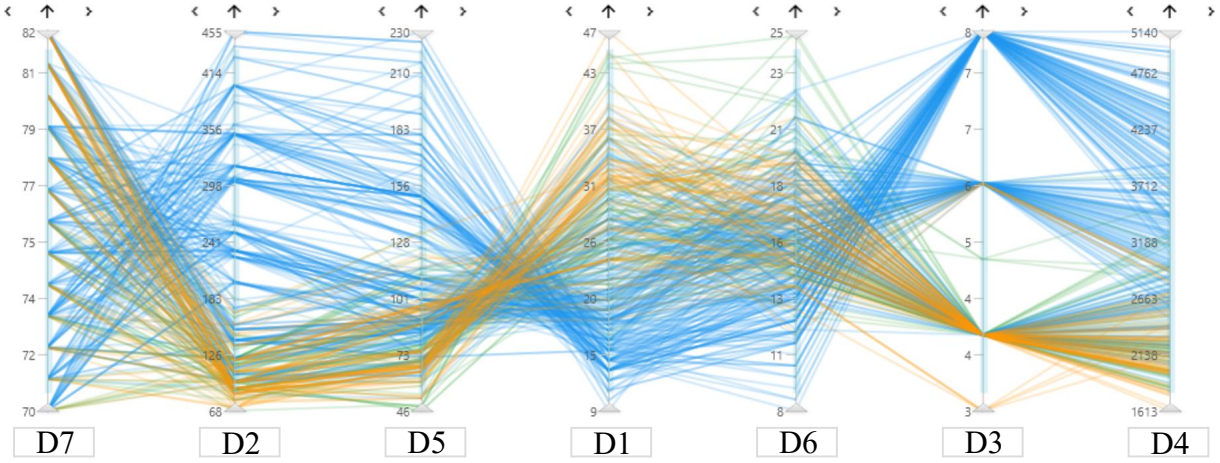
6.5.1.3 Automatic Reordering

In addition to visual guidance, the system reorders the axes upon request by the analyst. The system assigns a *score* to each axis, which represents the amount of time the user has spent on it. Netzel et al. [2017] found that users jump between axes while investigating parallel coordinates, rather than focusing on the areas between two axes. Therefore, in the sorting algorithm, looking at the axis itself has more impact on the *score* than spending time in the area between two axes. If the area between $axis_A$ and $axis_B$ is called $AREA_{A,B}$, the *score* assigned to $axis_A$ in timestamp t is calculated as: $score(axis_A, t) = \alpha \times f(axis_A, t) + f(AREA_{A,B}) + f(AREA_{A,C})$ where $axis_B$ and $axis_C$ are neighbour axes of $axis_A$, f is the amount of time spent on an AOI, and α is a coefficient.

The user can ask the system to sort any number of axes. The axes can be shown either from left to right, right to the left, or centre to side. Netzel et al. [2017] demonstrate how analysts are biased toward



(a) Before reordering, with the (usually hidden) internal eye-gaze heatmap.



(b) After reordering axes from centre to side.

Figure 6.12: (a) The analyst exploring a parallel coordinates plot of the well-known cars dataset. (b) Based on eye-gaze information captured during exploration, the system suggests a new ordering of the axes. In this variant, unexplored dimensions are pushed to the centre of the plot.

the centre and pay less attention to the sides in parallel coordinates. Therefore, the default is to sort the axes from centre to side. The analyst can decide whether to sort the axes in descending or ascending order. The former is called *organisation*, and the latter *exploration*. In organisation mode, the focus is on pushing insignificant axes to the side, and in exploration mode, the system aims to show unexplored information to the user.

By pressing the *Reorder* button, the system plays an animation to show the new location of the moving axis. Later, the axis will move to the new place, and the other axes will shift. So as not to overwhelm the user, only two axes are moved in each step. The technique is implemented using Java. For gaze input, an inexpensive EyeTribe tracker placed under the display, and configured to capture data at 30 Hz. The distance between the device and a participant is 50 cm. The application is run on a PC running Windows 10 and a 25-inch display with a resolution of 2560×1440 pixels and a frame rate of 60 Hz. The application uses Google Cloud Speech-to-Text API for speech recognition.

6.5.2 Eye-Tracking Visual Analysis Findings

The working prototype shows that using gaze information, based on AOI, can be an appropriate tool to interact with visualisation of multivariate datasets. This finding is proof to Hypothesis 5. In the future, a formal user study can further test this hypothesis.

Chapter 7

Future Work

"Cheers to gambler that lost everything he owned,
Nothing was left, except whim of gambling once more."

[Rumi, Persian poet, 1207–1273.]

The directions for further research follow the three main directions covered in this thesis: (a) local patterns (Chapter 3), (b) interactive visual labelling (Chapters 4 and 5), and (c) multimodal interaction (Chapter 6).

7.1 Local Scatterplot Patterns

Chapter 3 of the thesis presented a technique to search for and explore local patterns in SPLOMs. The algorithm has some room for improvement.

The presented algorithm is designed for general use cases. Therefore, some false positive patterns are found. This is due to not having a clear definition of patterns for a specific dataset. These matches can be excluded later by manual parameter tuning or use of the relevance feedback module. However, in future, it would be desirable to incorporate guidance to help the user find hidden local patterns. As an example, active learning algorithms can be integrated into the relevance feedback module, instead of one-time feedback from the user. Also, the user might be able to deselect undesired patterns rather than selecting the relevant ones.

An underlying question of scatterplot similarity is how the perception of patterns in scatterplots by analysts can be modelled, and eventually described by descriptors. In Pandey et al. [2016], an experiment to assess how analysts describe specific patterns in scatterplots was presented, which found that these were not easy to model using Scagnostics features. The shape and model-based descriptors used in the proposed approach are one choice, but additional features could be defined (or even learned from training data) to describe patterns more compatible with user perception and interpretation.

Since the descriptors are parameters in the approach, additional ones can be added to the system in the future. It would be particularly interesting to learn which descriptors work better with which kinds of dataset. After experimenting with L1, L2, and Quadratic Form distance functions, L1 was found to be good enough for the chosen datasets, possibly because of the coarseness of the descriptor's grid. It is possible that for other datasets and use cases, other distance functions might work better. Thus, it could be interesting to add distance functions as a parameter.

Since a sliding-window approach is used, the search may return many possible positions and areas of similarity within a given scatterplot. A simple rectangle is currently used to highlight matching patterns.

It would be interesting to research more advanced visual representations of local matches in a scatterplot. Also, if the data is labelled, instead of using the sliding-window approach, algorithmic complexity could be reduced by only comparing clusters of records with the same label, especially when the scatterplots are dense. Conducting visual cluster separation in scatterplots would be another alternative to the sliding-window approach.

Finally, it would be interesting to develop ground truth and benchmark datasets to further compare algorithms for local pattern discovery in scatterplots, adding to existing benchmarks for global scatterplot features [Scherer et al. 2012].

7.2 Interactive Visual Labelling

Chapter 4 of the thesis presented mVis, a tool for interactive visual labelling of multivariate datasets. In mVis, partitions are coupled with related dimensions based on interactions of the user. Currently, the system captures interactions performed with mouse and keyboard. In future, several modalities including eye tracking and voice recognition might help the system to find relationships between dimensions and partitions.

Since the labelling process is performed iteratively, it might be beneficial to keep a history of all user interactions and operations. The user may wish to revisit earlier labelling decisions, and possibly update the alphabet and partitions. Providing a visual history of labelling provenance, and how to propagate changes to earlier labelling decisions is an interesting research topic for future work. This also raises the need for appropriate comparative visualisation techniques [Gleicher et al. 2011], to contrast the different selections.

The results of the pre-study verified that mVis is moving in the right direction, but still lacks some key features. The observations showed that both participants could work with mVis without the need for much assistance. It confirmed that mVis is easy-to-use and fast to learn. Nevertheless, both participants asked for an interactive help module, and one of the participants requested a guidance module. Using this module, the system will show the correlation between dimensions and partitions as extra information. Moreover, the system could generate an automatic description of visible patterns in the dataset and guide the user toward them.

It is planned to release mVis as standalone, cross-platform software. First, however, several critical features, including proper dataset import and export, history tracking, and selection of dimensions still have to be implemented. Moreover, further case studies need to be conducted, including possibly more detailed and longer-term case studies using the full MILC methodology.

Currently, mVis has two primary types of data visualisation, scatterplots and parallel coordinates, but further visualisations could be added. For example, maps could be added to view records with attributes representing geo-coordinates. Also, more sophisticated glyphs can be used in scatterplots to carry more information to the analyst. The most important panel to add is a table view to show the full set of attribute values of records in the dataset.

Finally, one possibility to provide an analyst with interesting initial views in order to start labelling would be to use Scagnostics or Pargnostics features [Behrisch et al. 2018] to guide the user to relevant views.

Regarding Chapter 5, while the findings of this comparative study are interesting, they also depend on a number of choices made and would merit further investigation. For the experiments, a number of settings were fixed, which could be varied as well. Three specific visualisations (similarity map, SPLOM with scatterplot, and parallel coordinates) were chosen and these were used individually for the labelling task. Many visual analytics systems provide multiple linked views and dynamic brushing. Indeed, mVis provides these features too, but they were not used in this study in order to simplify its design. Multiple

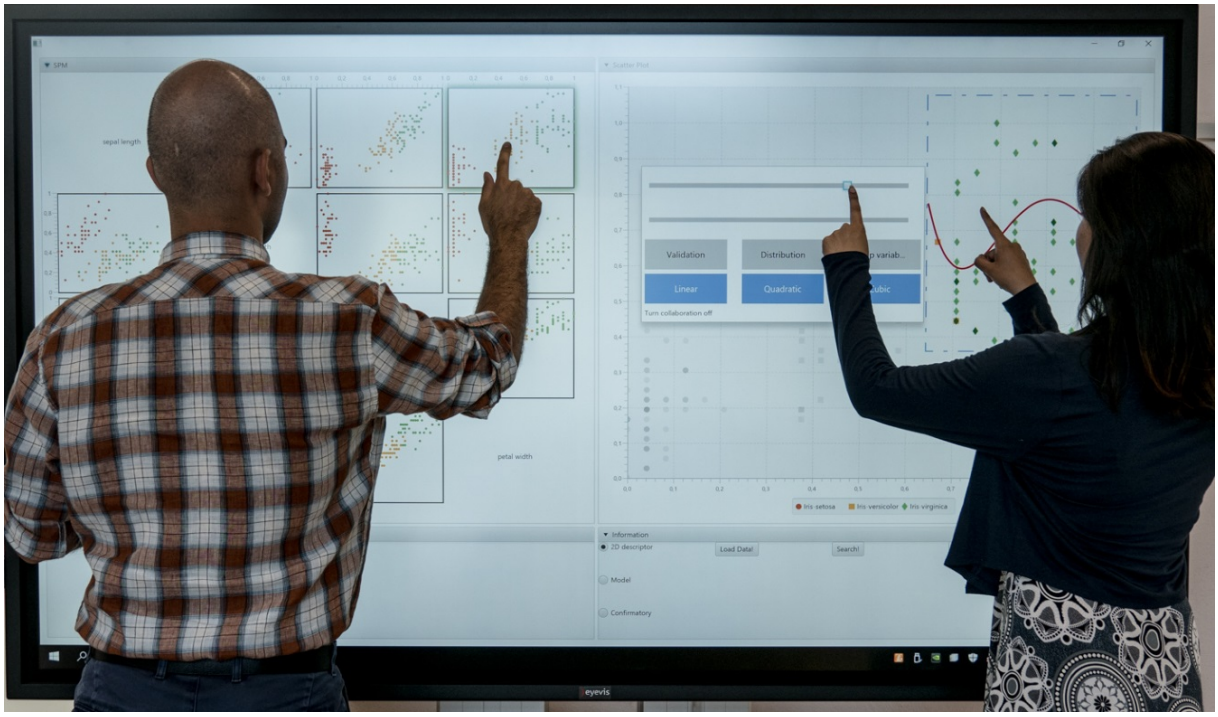


Figure 7.1: An example of solving visual analytics tasks on a large vertically-mounted multi-touch display.

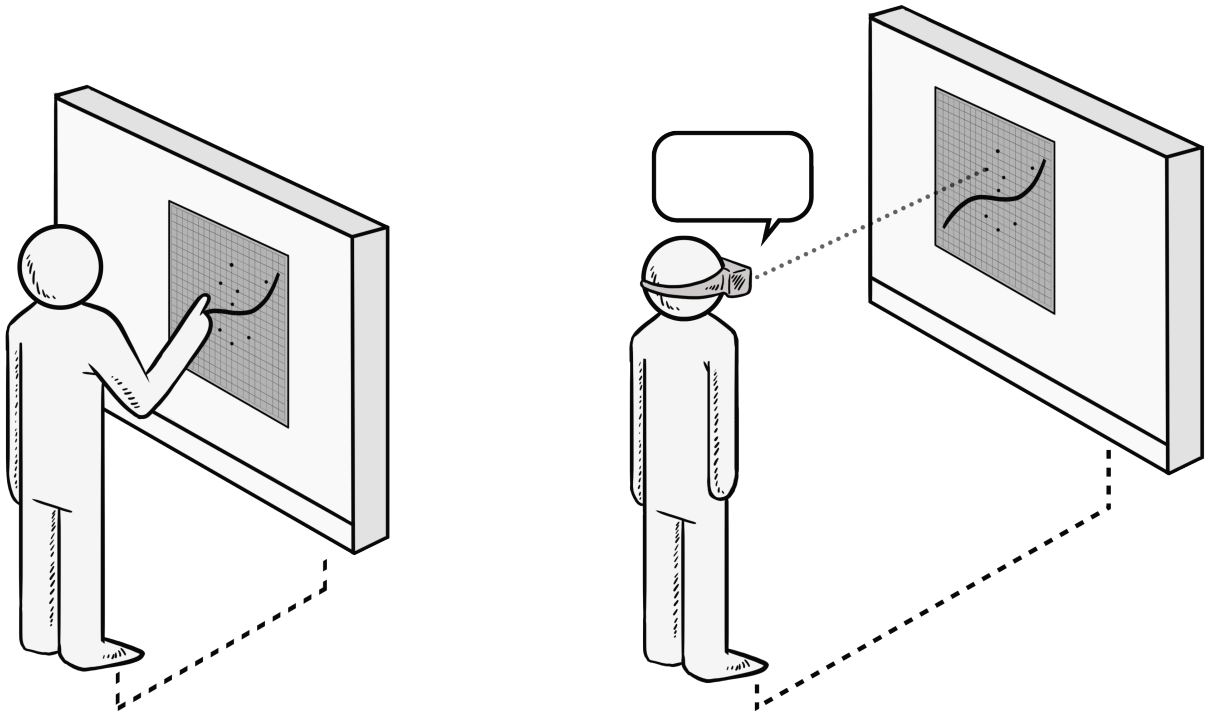
linked views and brushing could possibly mitigate some of the disadvantages of single techniques, and lead to a hierarchical selection strategy. For example, users might want to select a group of points as labelling candidates from the similarity view, and then switch to SPLOM or parallel coordinates for detailed selection and labelling. In future, support might be included for, say, automatic ordering of dimensions in parallel coordinates or arrangement of the plots in the SPLOM.

To compare classification performance, three different active learning algorithms (Smallest Margin, Entropy-Based Sampling, and Least Significant Confidence) were selected. While the selected algorithms are robust and applicable for different classifiers, the design space of active learning is large and more comparisons could be made.

In the accuracy comparison experiments, it is assumed that the user always assigns the true (ground truth) label for a data point, once it has been identified for labelling. While this corresponds to the notion of a user being an "oracle" in active learning, labelling errors could also be considered in a future experiment. Users could be allowed to freely pick a label, or even introduce a new label during interactive visual labelling. This would increase experimental complexity, but allow even more realistic assessments. Moreover, the analyst may want to assign multiple classes to a single record, or have a label alphabet with a heartache.

In many practical situations, the type and number of labels are not known in advance, but are determined in an iterative process. Also, in many practical problems, high-dimensional data attributes are often complemented with additional metadata and background information. For example, countries like in the WB dataset could be presented as map views. Including visualisation of such additional data, and studying how it is used during the labelling process, would be an interesting experiment to do.

In future work, it would be interesting to study the *dynamics* of the labelling process. For example, are there learning effects during labelling, where the choice of labels changes over time? In the experiment described in Chapter 5, the number of labels was fixed at 30. A future experiment could let the user decide when to stop the labelling process. To this end, feature and ML model space visualisations could



(a) Interacting with the screen using multi-touch (up close).

(b) Interacting with the screen using eye-tracking and speech (distance).

Figure 7.2: Two types of interaction with a large screen.

be helpful for the user to assess when label saturation has been reached.

7.3 Multimodal Interaction

Chapter 6 presented multiple ways for data interaction using novel devices. For future research, these modalities can be combined or improved for visual data analysis.

7.3.1 Large Multi-Touch Displays

Large vertically-mounted multi-touch displays, like the one shown in Figure 7.1, can be used to support collaborative visual analytics. A potential line of improvement is to adjust the view to the user's need and situation. In [Shao, Silva et al. 2017], the authors propose using eye tracking to infer user interest and using this information to recommend additional relevant but previously unseen views for exploration. While that work was developed as a desktop application, it might be interesting to incorporate eye-tracking support to recommend views for small collaborative team work on a large display. Moreover, adding group activity recognition and therefore pro-active interaction, can support collaboration by preventing information overload [Gordon et al. 2013].

Using only multi-touch for input can be overly restrictive. Other modalities need to be considered to utilise the power of these screens fully. By adding natural language interaction, the user can directly interact with the visual analytics application from a distance. Incorporating eye-tracking can help narrow down what the user is looking at or is interested in.

Figure 7.2a shows a setup of a large vertically-mounted multi-touch display. Due to the nature of touch screens, using multi-touch as the sole input modality can have some drawbacks, including the gorilla arm effect [Goodwins 2008], having to be within touching distance of the screen, and not being able to reach

all parts of the screen without stepping sideways. Figure 7.2b demonstrates an alternative scenario, in which the user is using eye-tracking and speech to interact with the scatterplot.

7.3.2 Second Handheld Device

Collaborative interaction for visual data analysis raises several interesting research challenges. For example, the server could track each members' data selection operations, and annotate them in the main display. The type of tracking and annotation is expected to depend on the analysis task. For example, when several experts collaborate to find local regression models in a given scatter plot, then each mobile display should show the current ML models proposed by the team members in a comparative way. Also, team members may interact with one another by passing and adapting each other's proposed regression models. Generally speaking, a collaborative visual analysis system should provide analysis provenance information, to allow comprehension of which operations have been performed by whom.

Another interesting problem is to adapt the display shown on the mobile devices to the respective device characteristics, such as display size and resolution. Depending on the device capabilities, the presentation and interaction operations should be tailored to fit by adapting the amount of data displayed and possibly changing the visualisation metaphor used (semantic zoom).

Chapter 8

Concluding Remarks

"From the depth of the black earth up to Saturn's apogee,
All the problems of the universe have been solved by me.
I have escaped from the coils of snares and deceits;
I have unravelled all knots except the knot of decease."

[Avicenna, Persian polymath, 980 - 1037.]

The novelty of the thesis is the presented line of research to bridge the gap between interaction modalities, and machine learning models for visual data analysis. Through the thesis, this pioneer research line is presented and validated by five hypotheses.

8.1 Résumé and Hypotheses

Here, the novelty of this thesis and validation of hypotheses are revisited.

Hypothesis 1 (H1): *Exploratory visual analytics, together with similarity search, is well suited for finding local patterns in scatterplot spaces.*

Chapter 3 presented a novel pipeline to search for local patterns in the scatterplots of a scatterplot matrix. Model-based and shape-based descriptors are used to compare the initial query pattern with other patterns. Relevance feedback is then used to refine the search. An implementation of the approach has shown its usefulness for various datasets and that it works in near real-time. Finally, the limitations and possible extensions of the approach were discussed.

Hypothesis 2 (H2): *By using interactive visualisation techniques, an analyst can build a machine learning ML model for a multivariate dataset.*

Chapter 4 demonstrated a new approach to make partitions on a multivariate dataset that does not contain any labelled record. Using appropriate views including partition similarity map, the analyst can manually label records with the help of classification, clustering and active learning algorithms. The result of the process is a properly labelled and partitioned dataset. An implementation of the approach called mVis had shown its usefulness for real-world datasets.

Hypothesis 3 (H3): *Interactive visual labelling techniques can surpass non-interactive labelling techniques based on active learning in terms of accuracy.*

Chapter 5 presented a study comparing three interactive visualisations with each other and with active learning for the purpose of labelling a multivariate dataset. The study also explored subjective user

ratings for the three interactive visualisations and discussed the labelling strategies employed by users with each them. All three interactive visualisations performed better than active learning algorithms, in terms of classification accuracy (assuming the user always assigns the correct label to a selected data instance). The similarity map performed better than both SPLOM with scatterplot and parallel coordinates in both the MNIST4 and WB datasets. Nevertheless, parallel coordinates and SPLOM with scatterplot are useful in their own right, especially for datasets where the dimensions have semantically meaningful names. The results support the view that a user-in-the-loop approach is beneficial for creating training datasets.

Hypothesis 4 (H4): *Large multi-touch displays facilitate collaborative analysis of ML models.*

Chapter 6 presented three novel techniques to use emerging interaction devices for data analysis. In the first section, it is demonstrated how two analysts can use a large multi-touch display collaboratively to analyse multivariate datasets. The same concept is applied in the second section, but multiple devices, including a handheld device is used for data analysis using scatterplots, and SPLOM.

Hypothesis 5 (H5): *Indirect feedback from gaze can improve interaction and visual exploration of a multivariate dataset.*

The third section of Chapter 6, presents a novel interaction technique with an inexpensive eye-tracker to re-arrange axes in a parallel coordinates.

8.2 Epilogue

When I started my doctoral degree, the title of my proposal did not contain the term *machine learning*. The focus was initially on data analysis using novel interaction modalities, such as large multi-touch displays, and eye-trackers. During the large multi-touch display project, a research gap between interactions with novel devices, and training ML models was found. Therefore, the focus shifted to create VA techniques that are not merely useful for data exploration, but for more advanced interactions with ML models. Although a small number of workshops have been held in recent years to address the gap between ML and interaction modalities using VA, the topic is still in its infancy. I hope by reading this thesis, researchers can come up with exciting new ideas.

There follows a list of publications written while doing this PhD, which were published in international peer-reviewed journals, conferences, and workshops.

Publications (9 Entries)

Berger, Philip, Mohammad Chegini, Heidrun Schumann and Christian Tominski [2018]. *Integrated Visualization of Structure and Attribute Similarity of Multivariate Graphs*. Poster at IEEE Conference on Information Visualization (InfoVis). Berlin, Germany, 2018 (cited on page 41).

Chegini, Mohammad, Keith Andrews, Tobias Schreck and Alexei Sourin [2019a]. *Eye-Tracking Based Adaptive Parallel Coordinates*. Proc. 2019 SIGGRAPH Asia Poster (SA'2019) (Brisbane, Australia). ACM, 17th Nov 2019, 44:1–44:2. ISBN 145036943X. doi:10.1145/3355056.3364563 (cited on page 7).

Chegini, Mohammad, Keith Andrews, Tobias Schreck and Alexei Sourin [2019b]. *Multiple Linked-View Exploration on Large Displays Facilitated by a Secondary Handheld Device*. International Workshop on Advanced Image Technology (IWAIT) 2019. Volume 11049. International Society for Optics and Photonics. 2019, 110490H (cited on page 7).

Chegini, Mohammad, Jürgen Bernard, Philip Berger, Alexei Sourin, Keith Andrews and Tobias Schreck [2019]. *Interactive Labelling of a Multivariate Dataset for Supervised Machine Learning Using Linked Visualisations, Clustering, and Active Learning*. Visual Informatics 3.1 (2019), pages 9–17. doi:10.1016/j.visinf.2019.03.002 (cited on page 6).

Chegini, Mohammad, Jürgen Bernard, Jian Cui, Fatemeh Chegini, Alexei Sourin, Keith Andrews and Tobias Schreck [2020]. *Interactive Visual Labelling versus Active Learning: An Experimental Comparison*. Frontiers of Information Technology & Electronic Engineering 21.4 (2020), pages 524–535. ISSN 2095-9184. doi:10.1631/FITEE.1900549 (cited on page 6).

Chegini, Mohammad, Jürgen Bernard, Lin Shao, Alexei Sourin, Keith Andrews and Tobias Schreck [2019]. *mVis in the Wild: Pre-Study of an Interactive Visual Machine Learning System for Labelling*. Proc. IEEE Vis 2019 Workshop on Evaluation of Interactive Visual Machine Learning Systems (EVIVA-ML). 2019 (cited on page 6).

Chegini, Mohammad, Lin Shao, Keith Andrews, Dirk J. Lehmann and Tobias Schreck [2017]. *Interaction Concepts for Collaborative Visual Analysis of Scatterplots on Large Vertically-Mounted High-Resolution Multi-Touch Displays*. Proc. Forum Media Technology (FMT 2017). 2017, pages 90–96 (cited on page 7).

Chegini, Mohammad, Lin Shao, Keith Andrews and Tobias Schreck [2018]. *Toward Multimodal Interaction of Scatterplot Spaces Exploration*. Proc. AVI Workshop on Multimodal Interaction for Data Visualization. 2018 (cited on page 7).

Chegini, Mohammad, Lin Shao, Robert Gregor, Dirk J. Lehmann, Keith Andrews and Tobias Schreck [2018]. *Interactive Visual Exploration of Local Patterns in Large Scatterplot Spaces*. Computer Graphics Forum (CGF) 37.3 (2018), pages 99–109 (cited on page 5).

Bibliography (148 Entries)

Amershi, Saleema, Maya Cakmak, W. Bradley Knox and Todd Kulesza [2014]. *Power to the People: The Role of Humans in Interactive Machine Learning*. AI Magazine 35.4 (2014), pages 105–120. doi:10.1609/aimag.v35i4.2513 (cited on page 32).

Andrews, Christopher, Alex Endert and Chris North [2010]. *Space to Think: Large High-Resolution Displays for Sensemaking*. Proceedings of the SIGCHI conference on human factors in computing systems. ACM. 2010, pages 55–64 (cited on page 62).

Andrews, Christopher, Alex Endert, Beth Yost and Chris North [2011]. *Information Visualization on Large, High-Resolution Displays: Issues, Challenges, and Opportunities*. Information Visualization 10.4 (2011), pages 341–355 (cited on page 15).

Andrews, Keith [2006]. *Evaluating Information Visualisations*. Proc. AVI 2006 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV'06). Venice, Italy: ACM Press, 2006, pages 1–5. ISBN 1595935622. doi:10.1145/1168149.1168151. http://ftp.isds.tugraz.at/pub/papers/andrews-beliv2006.pdf (cited on pages 13, 42).

Andrews, Keith [2008]. *Evaluation Comes in Many Guises*. Prof. CHI 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV'08). Florence, Italy, 2008. http://ftp.isds.tugraz.at/pub/papers/andrews-beliv08.pdf (cited on pages 13, 42).

Andrienko, Gennady, Natalia Andrienko, Salvatore Rinzivillo, Mirco Nanni, Dino Pedreschi and Fosca Giannotti [2009]. *Interactive Visual Clustering of Large Collections of Trajectories*. Proc. 2009 IEEE Symposium on Visual Analytics Science and Technology. IEEE. 2009, pages 3–10. doi:10.1109/VAST.2009.5332584 (cited on page 12).

Ashdown, Mark, Philip Tuddenham and Peter Robinson [2010]. *High-Resolution Interactive Displays*. In: *Tabletops - Horizontal Interactive Displays*. 2010, pages 71–100 (cited on page 63).

Attenberg, Josh and Foster Provost [2011]. *Inactive Learning?: Difficulties Employing Active Learning in Practice*. SIGKDD Explor. Newsl. 12.2 (2011), pages 36–41. ISSN 1931-0145. doi:10.1145/1964897.1964906 (cited on pages 32, 59).

Badam, Sriram Karthik, Fereshteh Amini, Niklas Elmqvist and Pourang Irani [2016]. *Supporting Visual Exploration for Multiple Users in Large Display Environments*. Visual Analytics Science and Technology (VAST), 2016 IEEE Conference on. IEEE. 2016, pages 1–10 (cited on page 15).

Beecks, Christian, Merih Seran Uysal and Thomas Seidl [2010]. *Signature Quadratic Form Distance*. Proc. ACM International Conference on Image and Video Retrieval (CIVR 2010) (Xi'an, China). 2010, pages 438–445. doi:10.1145/1816041.1816105 (cited on page 17).

Behrisch, M., M. Blumenschein, N. W. Kim, L. Shao, M. El-Assady, J. Fuchs, D. Seebacher, A. Diehl, U. Brandes, H. Pfister, T. Schreck, D. Weiskopf and D. A. Keim [2018]. *Quality Metrics for Information Visualization*. Computer Graphics Forum (EuroVis State of The Art Report) 37.3 (2018), pages 625–662. doi:10.1111/cgf.13446 (cited on page 78).

Bernard, Jürgen, Marco Hutter, Matthias Zeppelzauer, Dieter Fellner and Michael Sedlmair [2018]. *Comparing Visual-Interactive Labeling with Active Learning: An Experimental Study*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 24.1 (2018), pages 298–308. ISSN 1077-2626. doi:10.1109/TVCG.2017.2744818 (cited on pages 12, 31, 46–47, 49–50, 55).

Bernard, Jürgen, David Sessler, Tobias Ruppert, James Davey, Arjan Kuijper and Jörn Kohlhammer [2014]. *User-Based Visual-Interactive Similarity Definition for Mixed Data Objects-Concept and First Implementation*. Journal of WSCG 22 (2014), pages 329–338. ISSN 978-80-86943-71-8 (cited on page 12).

Bernard, Jürgen, Matthias Zeppelzauer, Markus Lehmann, Martin Müller and Michael Sedlmair [2018]. *Towards User-Centered Active Learning Algorithms*. Computer Graphics Forum (CGF) 37.3 (2018), pages 121–132. ISSN 1467-8659. doi:10.1111/cgf.13406 (cited on page 12).

Bernard, Jürgen, Matthias Zeppelzauer, Michael Sedlmair and Wolfgang Aigner [2018]. *VIAL: A Unified Process for Visual Interactive Labeling*. The Visual Computer 34.9 (2018), pages 1189–1207. ISSN 1432-2315. doi:10.1007/s00371-018-1500-3 (cited on pages 1, 12, 32, 47).

Bishop, Christopher M. [2006]. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 978-0387-31073-2 (cited on page 47).

Bruneau, Pierrick, Philippe Pinheiro, Bertjan Broeksema and Benoît Otjacques [2015]. *Cluster Sculptor, an Interactive Visual Clustering System*. Neurocomputing 150 (2015), pages 627–644. ISSN 0925-2312. doi:10.1016/j.neucom.2014.09.062 (cited on page 11).

Buja, A., J. A. McDonald, J. Michalak and W. Stuetzle [1991]. *Interactive Data Visualization Using Focusing and Linking*. Proc. 1991 on Visualization. IEEE. 1991, pages 156–163. doi:10.1109/VISUAL.1991.175794 (cited on page 70).

Card, Stuart K., Jock D. Mackinlay and Ben Shneiderman [1999]. *Readings in Information Visualization: Using Vision to Think*. San Francisco, CA, USA: Morgan Kaufmann, 1999. ISBN 1-55860-533-9 (cited on pages 1–2, 61).

Carpendale, Sheelagh [2008]. *Evaluating Information Visualizations*. In: *Information Visualization: Human-Centered Issues and Perspectives*. Edited by Andreas Kerren, John T. Stasko, Jean-Daniel Fekete and Chris North. LNCS 4950. Springer, 2008, pages 19–45. ISBN 354070955X. doi:10.1007/978-3-540-70956-5_2 (cited on pages 13, 42).

Ceneda, Davide, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jörg Schulz, Marc Streit and Christian Tominski [2016]. *Characterizing Guidance in Visual Analytics*. IEEE Transactions on Visualization and Computer Graphics 23.1 (2016), pages 111–120. doi:10.1109/TVCG.2016.2598468 (cited on pages 59, 73).

Chan, Yu-Hsuan, Carlos D. Correa and Kwan-Liu Ma [2010]. *Flow-Based Scatterplots for Sensitivity Analysis*. Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST 2010). 2010, pages 43–50. doi:10.1109/VAST.2010.5652460 (cited on page 9).

Chen, Haidong, Wei Chen, Honghui Mei, Zhiqi Liu, Kun Zhou, Weifeng Chen, Wentao Gu and Kwan-Liu Ma [2014]. *Visual Abstraction and Exploration of Multi-Class Scatterplots*. IEEE Transactions on Visualization and Computer Graphics 20.12 (2014), pages 1683–1692. doi:10.1109/TVCG.2014.2346594 (cited on page 9).

Choo, Jaegul, Hanseung Lee, Jaeyeon Kihm and Haesun Park [2010]. *iVisClassifier: An Interactive Visual Analytics System for Classification Based on Supervised Dimension Reduction*. Proc. 2010 IEEE Conference on Visual Analytics Science and Technology (VAST 2010). IEEE. 2010, pages 27–34. doi:10.1109/VAST.2010.5652443 (cited on pages 11–12, 31).

Cleveland, William S. [1993]. *Visualizing Data*. Hobart Press, 1993. ISBN 0963488406 (cited on page 61).

Collins, Christopher, Natalia Andrienko, Tobias Schreck, Jing Yang, Jaegul Choo, Ulrich Engelke, Amit Jena and Tim Dwyer [2018]. *Guidance in the Human-Machine Analytics Process*. Visual Informatics 2.3 (2018), pages 166–180 (cited on page 13).

Cox, Michael A. A. and Trevor F. Cox [2008]. *Multidimensional Scaling*. In: *Handbook of Data Visualization*. Springer, 2008, pages 315–347. ISBN 3540330372. doi:10.1007/978-3-540-33037-0_14 (cited on page 36).

Crisan, Anamaria and Madison Elliott [2018]. *How to Evaluate an Evaluation Study? Comparing and Contrasting Practices in Vis with Those of Other Disciplines*. Proc. Vis 2018 Workshop on Evaluation and Beyond – Methodological Approaches for Visualization (BELIV 2018) (Berlin, Germany). IEEE, 2018, pages 28–36. doi:10.1109/BELIV.2018.8634420 (cited on pages 13, 42).

Culotta, Aron and Andrew McCallum [2005]. *Reducing Labeling Effort for Structured Prediction Tasks*. Prof. 2005 National Conference on Artificial intelligence (AAAI). AAAI Press, 2005, pages 746–751. ISBN 1-57735-236-x (cited on page 12).

Czerwinski, Mary, Greg Smith, Tim Regan, Brian Meyers, George G Robertson and Gary Starkweather [2003]. *Toward Characterizing the Productivity Benefits of Very Large Displays.* Interact. Volume 3. 2003, pages 9–16 (cited on page 14).

DMandML [2018]. Github Repository. 2018. https://github.com/TKnudsen/DMandML (cited on page 39).

Eisemann, Martin, Georgia Albuquerque and Marcus Magnor [2014]. *A Nested Hierarchy of Localized Scatterplots*. Proc. Conference on Graphics, Patterns and Images (SIBGRAPI 2014). 2014, pages 80–86. doi:10.1109/SIBGRAPI.2014.14 (cited on page 10).

Eisen, Michael B., Paul T. Spellman, Patrick O. Brown and David Botstein [1998]. *Cluster Analysis and Display of Genome-Wide Expression Patterns*. Proc. National Academy of Sciences 95.25 (1998), pages 14863–14868. doi:10.1073/pnas.95.25.14863 (cited on page 9).

Endert, Alexander, William Ribarsky, Cagatay Turkay, BL William Wong, Ian Nabney, I Díaz Blanco and Fabrice Rossi [2018]. *The State of the Art in Integrating Machine Learning into Visual Analytics*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 24.7 (2018), pages 2223–2237. doi:10.1109/TVCG.2017.2711030 (cited on page 11).

Ericsson, K. Anders and Herbert A. Simon [1984]. *Protocol Analysis: Verbal Reports as Data*. MIT Press, 1984 (cited on page 44).

eyevis [2018]. *EYE-LCD-8400-QHD-V2*. 13th Apr 2018. http://eyevis.de/en/products/lcd-solutions/4k-ultra-hd-lcd-monitors/84-inch-4k-uhd-lcd.html (cited on pages 25, 64).

Fetter, Mirko and David Bimamisa [2015]. *TUIOFX—Toolkit Support for the Development of JavaFX Applications for Interactive Tabletops*. Proc. International Conference Human-Computer Interaction (INTERACT 2015). Springer, 2015, pages 486–489. doi:10.1007/978-3-319-22723-8_44 (cited on pages 25, 67).

Fetter, Mirko, David Bimamisa and Tom Gross [2017]. *TUIOFX: A JavaFX Toolkit for Shared Interactive Surfaces*. Proceedings of the ACM on Human-Computer Interaction 1.1 (2017), page 10 (cited on page 67).

Friedman, Nir and Stuart Russell [1997]. *Image Segmentation in Video Sequences: A Probabilistic Approach*. Proc. 13th Conference on Uncertainty in Artificial Intelligence (UAI'97). Morgan Kaufmann, 1997, pages 175–181 (cited on page 9).

Gleicher, Michael, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D. Hansen and Jonathan C. Roberts [2011]. *Visual Comparison for Information Visualization*. Information Visualization 10.4 (2011), pages 289–309. ISSN 1473-8716. doi:10.1177/1473871611416549 (cited on pages 10, 25, 78).

Goodwins, Rupert [2008]. *Windows 7? No Arm in it*. 28th May 2008. http://zdnet.com/article/windows-7-no-arm-in-it/ (cited on pages 15, 80).

Gordon, Dawud, Jan-Hendrik Hanne, Martin Berchtold, Ali Asghar Nazari Shirehjini and Michael Beigl [2013]. *Towards Collaborative Group Activity Recognition Using Mobile Devices*. Mobile Networks and Applications 18.3 (2013), pages 326–340 (cited on page 80).

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann and Ian H. Witten [2009]. *The WEKA Data Mining Software: An Update*. ACM SIGKDD Explorations Newsletter 11.1 (2009), pages 10–18. doi:10.1145/1656274.1656278 (cited on page 49).

Harrower, Mark and Cynthia A Brewer [2003]. *ColorBrewer. org: An Online Tool for Selecting Colour Schemes for Maps*. The Cartographic Journal 40.1 (2003), pages 27–37. doi:10.1179/000870403235002042 (cited on page 40).

Heilig, Mathias, Stephan Huber, Mischa Demarmels and Harald Reiterer [2010]. *Scattertouch: a Multi Touch Rubber Sheet Scatter Plot Visualization for Co-Located Data Exploration*. ACM International Conference on Interactive Tabletops and Surfaces. ACM. 2010, pages 263–264 (cited on page 14).

Heimerl, Florian, Steffen Koch, Harald Bosch and Thomas Ertl [2012]. *Visual Classifier Training for Text Document Retrieval*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 18.12 (2012), pages 2839–2848. ISSN 1077-2626. doi:10.1109/TVCG.2012.277 (cited on page 12).

Höferlin, Benjamin, Rudolf Netzel, Markus Höferlin, Daniel Weiskopf and Gunther Heidemann [2012]. *Inter-Active Learning of Ad-Hoc Classifiers for Video Visual Analytics*. Proc. 2012 IEEE Conference on Visual Analytics Science and Technology (VAST). IEEE. 2012, pages 23–32. doi:10.1109/VAST.2012.6400492 (cited on page 12).

Hoi, Steven CH, Rong Jin and Michael R Lyu [2006]. *Large-Scale Text Categorization by Batch Mode Active Learning*. Proc. 2006 International Conference on World Wide Web. ACM. 2006, pages 633–642. doi:10.1145/1135777.1135870 (cited on page 12).

Hund, Michael, Dominic Böhm, Werner Sturm, Michael Sedlmair, Tobias Schreck, Torsten Ullrich, Daniel A Keim, Ljiljana Majnaric and Andreas Holzinger [2016]. *Visual Analytics for Concept Exploration in Subspaces of Patient Groups*. Brain Informatics 3.4 (2016), pages 233–247. ISSN 2198-4026. doi:10.1007/s40708-016-0043-5 (cited on page 41).

Inselberg, Alfred [1985]. *The Plane with Parallel Coordinates*. The Visual Computer 1.2 (1985), pages 69–91. ISSN 1432-2315. doi:10.1007/BF01898350 (cited on pages 2, 11, 48, 61–62).

Isenberg, Petra, Sheelegh Carpendale, Anastasia Bezerianos, Nathalie Henry and Jean-Daniel Fekete [2009]. *Coconuttrix: Collaborative Retrofitting for Information Visualization*. IEEE Computer Graphics and Applications 29.5 (2009), pages 44–57 (cited on page 62).

Isenberg, Petra, Pierre Dragicevic, Wesley Willett, Anastasia Bezerianos and Jean-Daniel Fekete [2013]. *Hybrid-Image Visualization for Large Viewing Environments*. IEEE transactions on visualization and computer graphics 19.12 (2013), pages 2346–2355 (cited on page 14).

Isenberg, Petra and Danyel Fisher [2009]. *Collaborative Brushing and Linking for Co-located Visual Analytics of Document Collections*. Computer Graphics Forum 28.3 (2009), pages 1031–1038. ISSN 14678659. doi:10.1111/j.1467-8659.2009.01444.x (cited on page 70).

Isenberg, Petra, Tobias Isenberg, Tobias Hesselmann, Bongshin Lee, Ulrich Von Zadow and Anthony Tang [2013]. *Data Visualization on Interactive Surfaces: A Research Agenda*. IEEE Computer Graphics and Applications 33.2 (2013), pages 16–24 (cited on page 15).

Jakobsen, Mikkel R and Kasper Hornbæk [2014]. *Up Close and Personal: Collaborative Work on a High-Resolution Multitouch Wall Display*. ACM Transactions on Computer-Human Interaction (TOCHI) 21.2 (2014), page 11 (cited on pages 15, 62–63).

Jolliffe, Ian [2002]. *Principal Component Analysis*. 2nd Edition. Springer, 2002. ISBN 978-0387954424 (cited on page 52).

Kaltenbrunner, Martin, Till Bovermann, Ross Bencina and Enrico Costanza [2005]. *TUIO: A Protocol for Table-Top Tangible User Interfaces*. Proc. of the The 6th Int'l Workshop on Gesture in Human-Computer Interaction and Simulation. 2005, pages 1–5 (cited on pages 25, 67).

Karypis, George, Eui-Hong Han and Vipin Kumar [1999]. *Chameleon: Hierarchical Clustering Using Dynamic Modeling*. IEEE Computer 32.8 (1999), pages 68–75. ISSN 0018-9162. doi:10.1109/2.781637 (cited on page 12).

Keim, D.A. [2002]. *Information Visualization and Visual Data Mining*. IEEE Transactions on Visualization and Computer Graphics 8.1 (2002), pages 1–8. ISSN 1077-2626. doi:10.1109/2945.981847 (cited on page 14).

Keim, Daniel, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer and Guy Melançon [2008]. *Visual Analytics: Definition, Process, and Challenges*. In: *Information Visualization: Human-Centered Issues and Perspectives*. Edited by Andreas Kerren, John T. Stasko, Jean-Daniel Fekete and Chris North. Springer, 2008, pages 154–175. ISBN 978-3-540-70955-8. doi:10.1007/978-3-540-70956-5_7 (cited on pages 1–3).

Khan, Taimur K [2011]. *A Survey of Interaction Techniques and Devices for Large High Resolution Displays*. OASIcs-OpenAccess Series in Informatics. Volume 19. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. 2011 (cited on pages 14, 62).

Kister, U, K Klamka, C Tominski and R Dachselt [2017]. *GraSp: Combining Spatially-aware Mobile Devices and a Display Wall for Graph Visualization and Interaction*. Computer Graphics Forum 36.3 (2017), pages 503–514. ISSN 14678659. doi:10.1111/cgf.13206 (cited on pages 15, 68).

Kister, Ulrike, Patrick Reipschläger and Raimund Dachselt [2016]. *MultiLens: Fluent Interaction with Multi-Functional Multi-Touch Lenses for Information Visualization*. Proceedings of the 2016 ACM on Interactive Surfaces and Spaces. ACM. 2016, pages 139–148 (cited on page 14).

Kister, Ulrike, Patrick Reipschläger, Fabrice Matulic and Raimund Dachselt [2015]. *BodyLenses: Embodied Magic Lenses and Personal Territories for Wall Displays*. Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces. ACM. 2015, pages 117–126 (cited on page 65).

Kottke, Daniel, Adrian Calma, Denis Huseljic, Georg Krempl and Bernhard Sick [2017]. *Challenges of Reliable, Realistic and Comparable Active Learning Evaluation*. Proc. Interactive Adaptive Learning Workshop. 2017, pages 1–14. http://www.daniel.kottke.eu/2017/challenges-of-reliable-realistic-and-comparable-active-learning-evaluation/ (cited on page 59).

Kruskal, Joseph B. [1964]. *Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis*. Psychometrika 29.1 (1964), pages 1–27. doi:10.1007/BF02289565 (cited on page 55).

Kwon, Bum Chul, Ben Eysenbach, Janu Verma, Kenney Ng, Christopher De Filippi, Walter F Stewart and Adam Perer [2018]. *Clustervision: Visual Supervision of Unsupervised Clustering*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 24.1 (2018), pages 142–151. ISSN 1077-2626. doi:10.1109/TVCG.2017.2745085 (cited on page 12).

Lam, Heidi, Enrico Bertini, Petra Isenberg, Catherine Plaisant and Sheelagh Carpendale [2012]. *Empirical Studies in Information Visualization: Seven Scenarios*. IEEE Transactions on Visualization and

Computer Graphics (TVCG) 18.9 (2012), pages 1520–1536. ISSN 1077-2626. doi:10.1109/TVCG.2011.279 (cited on pages 13, 42).

Langner, R., U. Kister and R. Dachselt [2018]. *Multiple Coordinated Views at Large Displays for Multiple Users: Empirical Findings on User Behavior, Movements, and Distances*. IEEE Transactions on Visualization and Computer Graphics (TVCG) Early Access (2018). ISSN 1077-2626. doi:10.1109/TVCG.2018.2865235 (cited on page 15).

LeCun, Yann, Léon Bottou, Yoshua Bengio and Patrick Haffner [1998]. *Gradient-Based Learning Applied to Document Recognition*. Proceedings of the IEEE 86.11 (1998), pages 2278–2324. doi:10.1109/5.726791 (cited on pages 49, 52).

Lee, Bongshin, Petra Isenberg, Nathalie Henry Riche and Sheelagh Carpendale [2012]. *Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 18.12 (2012), pages 2689–2698. ISSN 1077-2626. doi:10.1109/TVCG.2012.204 (cited on pages 2, 13, 61).

Lee, Bongshin, Arjun Srinivasan, John Stasko, Melanie Tory and Vidya Setlur [2018]. *Multimodal Interaction for Data Visualization*. Proc. 2018 International Conference on Advanced Visual Interfaces (AVI 2018). ACM. 2018, 11:1–11:3. doi:10.1145/3206505.3206602 (cited on pages 2, 61).

Lee, Hanseung, Jaeyeon Kihm, Jaegul Choo, John Stasko and Haesun Park [2012]. *iVisClustering: An Interactive Visual Document Clustering via Topic Modeling*. Computer Graphics Forum (CGF) 31.3pt3 (2012), pages 1155–1164. doi:10.1111/j.1467-8659.2012.03108.x (cited on page 11).

Lichman, M. [2013]. *UCI Machine Learning Repository*. 2013. `http://archive.ics.uci.edu/ml` (cited on page 67).

Lin, Hanfei, Siyuan Gao, David Gotz, Fan Du, Jingrui He and Nan Cao [2017]. *RCLens: Interactive Rare Category Exploration and Identification*. Computer Graphics Forum (CGF) 36.8 (2017), pages 458–486. doi:10.1111/cgf.13092 (cited on pages 11, 31).

Liu, Can, Olivier Chapuis, Michel Beaudouin-Lafon and Eric Lecolinet [2017]. *CoReach: Cooperative Gestures for Data Manipulation on Wall-Sized Displays*. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems. ACM. 2017, pages 6730–6741 (cited on pages 15, 62–63, 66).

Lloyd, Stuart [1982]. *Least Squares Quantization in PCM*. IEEE Transactions on Information Theory 28.2 (1982), pages 129–137. ISSN 0018-9448. doi:10.1109/TIT.1982.1056489 (cited on page 12).

Maaten, Laurens van der and Geoffrey Hinton [2008]. *Visualizing Data Using t-SNE*. Journal of Machine Learning Research 9.Nov (2008), pages 2579–2605. ISSN 1077-2626 (cited on page 49).

Mamitsuka, Naoki Abe Hiroshi [1998]. *Query Learning Strategies Using Boosting and Bagging*. Proc. 1998 International Conference on Machine Learning (ICML). Volume 1. Morgan Kaufmann. 1998, pages 1–9 (cited on page 12).

Mann, Richard, Allan D. Jepson and Thomas El-Maraghi [2002]. *Trajectory Segmentation Using Dynamic Programming*. Proc. 16th International Conference on Pattern Recognition. Volume 1. 2002, pages 331–334. doi:10.1109/ICPR.2002.1044709 (cited on page 9).

Matković, Krešimir, Hrvoje Abraham, Mario Jelović and Helwig Hauser [2017]. *Quantitative Externalization of Visual Data Analysis Results Using Local Regression Models*. International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Springer. 2017, pages 199–218 (cited on page 14).

Matute, José, Alexandru C. Telea and Lars Linsen [2018]. *Skeleton-Based Scagnosticss*. IEEE Transactions on Visualization and Computer Graphics 24.1 (2018), pages 542–552. doi:10.1109/TVCG.2017.2744339 (cited on page 10).

Mayorga, Adrian and Michael Gleicher [2013]. *Splatterplots: Overcoming Overdraw in Scatter Plots*. IEEE Transactions on Visualization and Computer Graphics 19.9 (2013), pages 1526–1538. doi:10.1109/TVCG.2013.65 (cited on page 9).

Morris, Meredith Ringel, Anqi Huang, Andreas Paepcke and Terry Winograd [2006]. *Cooperative Gestures: Multi-User Gestural Interactions for Co-Located Groupware*. Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM. 2006, pages 1201–1210 (cited on pages 15, 62).

Nam, Eun Ju, Yiping Han, Klaus Mueller, Alla Zelenyuk and Dan Imre [2007]. *ClusterSculptor: A Visual Analytics Tool for High-Dimensional Data*. Proc. 2007 IEEE Symposium on Visual Analytics Science and Technology. IEEE. 2007, pages 75–82. doi:10.1109/VAST.2007.4388999 (cited on page 12).

Netzel, Udolf, Jenny Vuong, Ulrich Engelke, Seán O'Donoghue, Daniel Weiskopf and Julian Heinrich [2017]. *Comparative Eye-Tracking Evaluation of Scatterplots and Parallel Coordinates*. Visual Informatics 1.2 (2017), pages 118–131. ISSN 2468-502X. doi:10.1016/j.visinf.2017.11.001 (cited on pages 2, 73).

Nhon, Dang Tuan, Anushka Anand and Leland Wilkinson [2013]. *TimeSeer: Scagnostics for High-Dimensional Time Series*. IEEE Transactions on Visualization and Computer Graphics 19.3 (2013), pages 470–483. doi:10.1109/TVCG.2012.128 (cited on page 10).

Nhon, Dang Tuan and Leland Wilkinson [2014]. *PixSearcher: Searching Similar Images in Large Image Collections through Pixel Descriptors*. Proc. International Symposium on Visual Computing (ISVC 2014). Volume 8888. LNCS. Springer, 2014, pages 726–735. doi:10.1007/978-3-319-14364-4_70 (cited on page 10).

Nilsson, Nils J [1965]. *Learning Machines.* (1965) (cited on page 2).

North, Chris [2006]. *Toward Measuring Visualization Insight*. IEEE Computer Graphics and Applications 26.3 (2006), pages 6–9. doi:10.1109/MCG.2006.70 (cited on pages 13, 42).

Paiva, Jose Gustavo, William Robson Schwartz, Helio Pedrini and Rosane Minghim [2015]. *An Approach to Supporting Incremental Visual Data Classification*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 21.1 (2015), pages 4–17. ISSN 1077-2626. doi:10.1109/TVCG.2014.2331979 (cited on page 12).

Pandey, Anshul Vikram, Josua Krause, Cristian Felix, Jeremy Boy and Enrico Bertini [2016]. *Towards Understanding Human Similarity Perception in the Analysis of Large Sets of Scatter Plots*. Proc. 2016 CHI Conference on Human Factors in Computing Systems. ACM. 2016, pages 3659–3669. doi:10.1145/2858036.2858155 (cited on pages 20, 27, 77).

Pedersen, Esben Warming and Kasper Hornbæk [2012]. *An Experimental Comparison of Touch Interaction on Vertical and Horizontal Surfaces*. Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design. ACM. 2012, pages 370–379 (cited on page 15).

Perer, Adam and Ben Shneiderman [2009]. *Integrating Statistics and Visualization for Exploratory Power: From Long-Term Case Studies to Design Guidelines*. IEEE Computer Graphics and Applications 29.3 (2009), pages 39–51. doi:10.1109/MCG.2009.44 (cited on pages 13, 42).

Plaisant, Catherine [2004]. *The Challenge of Information Visualization Evaluation*. Proc. Advanced Visual Interfaces (AVI 2004). Gallipoli, Italy: ACM, 2004, pages 109–116. ISBN 1581138679. doi:10.1145/989863.989880 (cited on pages 13, 42).

Prouzeau, Arnaud, Anastasia Bezerianos and Oliver Chapuis [2016a]. *Evaluating Multi-User Selection for Exploring Graph Topology on Wall-Displays*. IEEE Transactions on Visualization and Computer Graphics (2016) (cited on page 62).

Prouzeau, Arnaud, Anastasia Bezerianos and Olivier Chapuis [2016b]. *Towards Road Traffic Management with Forecasting on Wall Displays*. Proceedings of the 2016 ACM on Interactive Surfaces and Spaces. ACM. 2016, pages 119–128 (cited on page 65).

Prouzeau, Arnaud, Anastasia Bezerianos and Olivier Chapuis [2017]. *Trade-offs Between a Vertical Shared Display and Two Desktops in a Collaborative Path-Finding Task*. Proceedings of Graphics Interface 2017. 2017 (cited on page 15).

Qi, Guo-Jun, Xian-Sheng Hua, Yong Rui, Jinhui Tang and Hong-Jiang Zhang [2009]. *Two-dimensional Multilabel Active Learning with an Efficient Online Adaptation Model for Image Classification*. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 31.10 (2009), pages 1880–1897. ISSN 0162-8828. doi:10.1109/TPAMI.2008.218 (cited on page 12).

Rasmussen, Matt and George Karypis [2004]. *gCLUTO: An Interactive Clustering, Visualization, and Analysis System*. Tech. Report CSE/UMN TR 04-021. Univ. of Minnesota, Department of Computer Science and Engineering, CSE, 2004 (cited on page 12).

Reda, Khairi, Andrew E Johnson, Michael E Papka and Jason Leigh [2015]. *Effects of Display Size and Resolution on User Behavior and Insight Acquisition in Visual Exploration*. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems. ACM. 2015, pages 2759–2768 (cited on pages 62–63).

Ritter, Christian, Christian Altenhofen, Matthias Zeppelzauer, Arjan Kuijper, Tobias Schreck and Jürgen Bernard [2018]. *Personalized Visual-Interactive Music Classification*. Proc. 2018 EuroVis Workshop on Visual Analytics (EuroVA). Wiley, 2018. doi:10.2312/eurova.20181109 (cited on page 12).

Roberts, Jonathan C [2005]. *Exploratory Visualization with Multiple Linked Views*. In: *Exploring Geovisualization*. Amsterdam: Elseviers, 2005, pages 159–180 (cited on pages 1, 14).

Ruddle, Roy A, Waleed Fateen, Darren Treanor, Peter Sondergeld and Phil Ouirke [2013]. *Leveraging Wall-Sized High-Resolution Displays for Comparative Genomics Analyses of Copy Number Variation*. Biological Data Visualization (BioVis), 2013 IEEE Symposium on. IEEE. 2013, pages 89–96 (cited on page 62).

Ruddle, Roy A, Rhys G Thomas, Rebecca S Randell, Phil Quirke and Darren Treanor [2015]. *Performance and Interaction Behaviour During Visual Search on Large, High-Resolution Displays*. Information Visualization 14.2 (2015), pages 137–147 (cited on page 14).

Rzeszotarski, Jeffrey M and Aniket Kittur [2014]. *Kinetica: Naturalistic Multi-Touch Data Visualization*. Proceedings of the 32nd annual ACM conference on Human factors in computing systems. ACM. 2014, pages 897–906 (cited on page 14).

Sacha, Dominik, Michael Sedlmair, Leishi Zhang, John A. Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North and Daniel A. Keim [2017]. *What You See is What You Can Change: Human-Centered Machine Learning by Interactive Visualization*. Neurocomputing 268 (2017), pages 164–175. ISSN 0925-2312. doi:10.1016/j.neucom.2017.01.105 (cited on page 32).

Sacha, Dominik, Leishi Zhang, Michael Sedlmair, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North and Daniel A Keim [2017]. *Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis*. IEEE Transactions on Visualization and Computer Graphics 23.1 (2017), pages 241–250. ISSN 1077-2626. doi:10.1109/TVCG.2016.2598495 (cited on page 11).

Sadana, Ramik and John Stasko [2016]. *Expanding Selection for Information Visualization Systems on Tablet Devices*. Proceedings of the 2016 ACM on Interactive Surfaces and Spaces. ACM. 2016, pages 149–158 (cited on page 14).

Samuel, A. L. [1959]. *Some Studies in Machine Learning Using the Game of Checkers*. IBM Journal of Research and Development 3.3 (1959), pages 210–229. ISSN 0018-8646. doi:10.1147/rd.33.0210 (cited on page 1).

Sarikaya, Alper and Michael Gleicher [2018]. *Scatterplots: Tasks, Data, and Designs*. IEEE Transactions on Visualization and Computer Graphics 24.1 (2018), pages 402–412. doi:10.1109/TVCG.2017.2744184 (cited on page 4).

Scheffer, Tobias, Christian Decomain and Stefan Wrobel [2001]. *Active Hidden Markov Models for Information Extraction*. Proc. 2001 International Conference on Advances in Intelligent Data Analysis (IDA). Springer, 2001, pages 309–318. ISBN 3-540-42581-0. doi:10.1007/3-540-44816-0_31 (cited on page 12).

Scherer, Maximilian, Jürgen Bernard and Tobias Schreck [2011]. *Retrieval and Exploratory Search in Multivariate Research Data Repositories Using Regressional Features*. Proc. ACM/IEEE Joint Conference on Digital Libraries (JCDL'11). ACM, 2011, pages 363–372. doi:10.1145/1998076.1998144 (cited on page 10).

Scherer, Maximilian, Tatiana von Landesberger and Tobias Schreck [2012]. *A Benchmark for Content-Based Retrieval in Bivariate Data Collections*. Proc. 2nd International Conference on Theory and Practice of Digital Libraries. Volume 7489. LNCS. 2012, pages 286–297. doi:10.1007/978-3-642-33290-6_31 (cited on pages 10, 78).

Scherer, Maximilian, Tatiana von Landesberger and Tobias Schreck [2013]. *Visual-Interactive Querying for Multivariate Research Data Repositories Using Bag-of-Words*. Proc. 13th ACM/IEEE Joint Conference on Digital Libraries (Indianapolis, Indiana, USA). 2013, pages 285–294. doi:10.1145/2467696.2467705 (cited on page 10).

Schreck, Tobias and Christian Panse [2007]. *A New Metaphor for Projection-based Visual Analysis and Data Exploration*. Proc. Electronic Imaging Conference on Visualization and Data Analysis. Volume 6495. SPIE. 2007. doi:10.1117/12.697879 (cited on page 10).

Schreck, Tobias, Tatiana von Landesberger and Sebastian Bremm [2010]. *Techniques for Precision-Based Visual Analysis of Projected Data*. Information Visualization 9.3 (2010), pages 181–193. doi:10.1057/ivs.2010.2 (cited on page 55).

Sedlmair, Michael, Miriah Meyer and Tamara Munzner [2012]. *Design Study Methodology: Reflections from the Trenches and the Stacks*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 18.12 (2012), pages 2431–2440. ISSN 1077-2626. doi:10.1109/TVCG.2012.213 (cited on page 13).

Sedlmair, Michael, Andrada Tatu, Tamara Munzner and Melanie Tory [2012]. *A Taxonomy of Visual Cluster Separation Factors*. Computer Graphics Forum 31.3pt4 (2012), pages 1335–1344. doi:10.1111/j.1467-8659.2012.03125.x (cited on page 9).

Sedlmair, Michael, Andrada Tatu, Tamara Munzner and Melanie Tory [2015]. *Data-Driven Evaluation of Visual Quality Measures*. Computer Graphics Forum 34.3 (2015), pages 201–210. doi:10.1111/cgf.12632 (cited on page 9).

Settles, Burr [2009]. *Active Learning Literature Survey*. Tech. Report 1648. Univ. of Wisconsin–Madison, 2009 (cited on page 47).

Settles, Burr [2012]. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning 6.1 (2012), pages 1–114. doi:10.2200/S00429ED1V01Y201207AIM018 (cited on pages 3, 11).

Settles, Burr and Mark Craven [2008]. *An Analysis of Active Learning Strategies for Sequence Labeling Tasks*. Proc. 2008 Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics. 2008, pages 1070–1079 (cited on page 12).

Seung, H. S., M. Opper and H. Sompolinsky [1992]. *Query by Committee*. Proc. 1992 Workshop on Computer Learning Theory (COLT). ACM. 1992, pages 287–294. doi:10.1145/130385.130417 (cited on page 12).

Shao, Lin, Michael Behrisch, Tobias Schreck, Tatiana von Landesberger, Maximilian Scherer, Sebastian Bremm and Daniel A. Keim [2014]. *Guided Sketching for Visual Search and Exploration in Large Scatter Plot Spaces*. Proc. EuroVis Workshop on Visual Analytics (EuroVA 2014). 2014. doi:10.2312/eurova.20141140 (cited on pages 10, 25).

Shao, Lin, Aishwarya Mahajan, Tobias Schreck and Dirk J. Lehmann [2017]. *Interactive Regression Lens for Exploring Scatter Plots*. Computer Graphics Forum (CGF) 36.3 (2017), pages 157–166. doi:10.1111/cgf.13176 (cited on pages 1, 10, 14, 17, 25, 37, 63–64, 70).

Shao, Lin, Timo Schleicher, Michael Behrisch, Tobias Schreck, Ivan Sipiran and Daniel A. Keim [2016]. *Guiding the Exploration of Scatter Plot Data Using Motif-Based Interest Measures*. Journal of Visual Languages & Computing 36 (2016), pages 1–12. doi:10.1016/j.jvlc.2016.07.003 (cited on page 9).

Shao, Lin, Nelson Silva, Eva Eggeling and Tobias Schreck [2017]. *Visual Exploration of Large Scatter Plot Matrices by Pattern Recommendation Based on Eye Tracking*. Proc. ACM Workshop on Exploratory Search and Interactive Data Analytics. 2017, pages 9–16. doi:10.1145/3038462.3038463 (cited on pages 14, 73, 80).

Shneiderman, Ben and Catherine Plaisant [2006]. *Strategies for Evaluating Information Visualization Tools: Multi-dimensional In-depth Long-term Case Studies*. Proc. AVI 2006 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV '06). Venice, Italy: ACM Press, 2006, pages 1–7. ISBN 1595935622. doi:10.1145/1168149.1168158 (cited on pages 13, 42).

Srinivasan, Arjun and John Stasko [2018]. *Orko: Facilitating Multimodal Interaction for Visual Exploration and Analysis of Networks*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 24.1 (2018), pages 511–521. ISSN 1077-2626. doi:10.1109/TVCG.2017.2745219 (cited on pages 14, 68).

Tatu, Andrada, Fabian Maaß, Ines Färber, Enrico Bertini, Tobias Schreck, Thomas Seidl and Daniel Keim [2012]. *Subspace Search and Visualization to Make Sense of Alternative Clusterings in High-Dimensional Data*. Proc. IEEE Conference on Visual Analytics Science and Technology (VAST 2012). 2012, pages 63–72. doi:10.1109/VAST.2012.6400488 (cited on pages 1, 10).

Tominski, Christian, Camilla Forsell and Jimmy Johansson [2012]. *Interaction Support for Visual Comparison Inspired by Natural Behavior*. IEEE Transactions on Visualization and Computer Graphics 18.12 (2012), pages 2719–2728. doi:10.1109/TVCG.2012.237 (cited on page 10).

Tsandilas, Theophanis, Anastasia Bezerianos and Thibaut Jacob [2015]. *SketchSliders: Sketching Widgets for Visual Exploration on Wall Displays*. Proc. 2015 Conference on Human Factors in Computing Systems (ACM CHI'15). ACM. 2015, pages 3255–3264. doi:10.1145/2702123.2702129 (cited on pages 14, 68).

Tuia, Devis, Michele Volpi, Loris Copa, Mikhail Kanevski and Jordi Munoz-Mari [2011]. *A Survey of Active Learning Algorithms for Supervised Remote Sensing Image Classification*. IEEE Journal of Selected Topics in Signal Processing 5.3 (2011), pages 606–617. ISSN 1932-4553. doi:10.1109/JSTSP.2011.2139193 (cited on page 12).

Turk, Matthew [2014]. *Multimodal Interaction: A Review*. Pattern Recognition Letters 36 (2014), pages 189–195 (cited on page 13).

TWB [2018]. *Countries and Economies*. The World Bank Group. Apr 2018. https://data.worldbank.org/country (cited on pages 22, 28, 52).

Valiati, Eliane R. A., Carla M. D. S. Freitas and Marcelo S. Pimenta [2008]. *Using Multi-Dimensional In-Depth Long-Term Case Studies for Information Visualization Evaluation*. Proc. CHI 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV 2008). Florence, Italy: ACM, 2008, pages 1–7. doi:10.1145/1377966.1377978 (cited on page 13).

Vogt, Katherine, Lauren Bradel, Christopher Andrews, Chris North, Alex Endert and Duke Hutchings [2011]. *Co-Located Collaborative Sensemaking on a Large High-Resolution Display with Multiple Input Devices*. Human-Computer Interaction–INTERACT 2011 (2011), pages 589–604 (cited on page 62).

Wang, Jian [1995]. *Integration of Eye-Gaze, Voice and Manual Response in Multimodal User Interface*. Proc. IEEE International Conference on Systems, Man and Cybernetics. Volume 5. Oct 1995, pages 3938–3942. doi:10.1109/ICSMC.1995.538404 (cited on page 14).

Wenskovitch, John, Ian Crandell, Naren Ramakrishnan, Leanna House and Chris North [2018]. *Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics*. IEEE Transactions on Visualization and Computer Graphics (TVCG) 24.1 (2018), pages 131–141. doi:10.1109/TVCG.2017.2745258 (cited on pages 2, 11).

Wilkinson, Leland, Anushka Anand and Robert Grossman [2005]. *Graph-Theoretic Scagnostics*. Proc. IEEE Symposium on Information Visualization (InfoVis 2005). 2005, pages 157–164. doi:10.1109/INFVIS.2005.1532142 (cited on page 10).

Wong, Yuet Ling, Krishna Madhavan and Niklas Elmqvist [2018]. *Towards Characterizing Domain Experts as a User Group*. Proc. IEEE Evaluation and Beyond - Methodological Approaches for Visualization (BELIV 2018). IEEE, 2018, pages 1–10. doi:10.1109/BELIV.2018.8634026 (cited on page 13).

Wu, Yi, Igor Kozintsev, Jean-Yves Bouguet and Carole Dulong [2006]. *Sampling Strategies for Active Learning in Personal Photo Retrieval*. Proc. 2006 IEEE International Conference on Multimedia and Expo. IEEE, 2006, pages 529–532. doi:10.1109/ICME.2006.262442 (cited on page 12).

Yates, Andrew, Allison Webb, Michael Sharpnack, Helen Chamberlin, Kun Huang and Raghu Machiraju [2014]. *Visualizing Multidimensional Data with Glyph SPLOMs*. Computer Graphics Forum 33.3 (2014), pages 301–310. doi:10.1111/cgf.12386 (cited on page 10).

Yee, Ka-Ping [2004]. *Two-Handed Interaction on a Tablet Display*. CHI'04 Extended Abstracts on Human Factors in Computing Systems. ACM. 2004, pages 1493–1496 (cited on page 65).

Yi, Ji Soo, Youn Ah Kang, John Stasko and Julie Jacko [2007]. *Toward a Deeper Understanding of the Role of Interaction in Information Visualization*. IEEE transactions on visualization and computer graphics 13.6 (2007), pages 1224–31. ISSN 1077-2626. doi:10.1109/TVCG.2007.70515 (cited on page 13).

Yu, Hualong, Changyin Sun, Wankou Yang, Xibei Yang and Xin Zuo [2015]. *AL-ELM: One Ucertainty-Based Active Learning Algorithm Using Extreme Learning Machine*. Neurocomputing 166 (2015), pages 140–150. ISSN 0925-2312. doi:10.1016/j.neucom.2015.04.019 (cited on page 4).

Zhai, Yan, Guoying Zhao, Toni Alatalo, Janne Heikkilä, Timo Ojala and Xinyuan Huang [2013]. *Gesture Interaction for Wall-Sized Touchscreen Display*. Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication. ACM. 2013, pages 175–178 (cited on page 14).