Stony Brook University

VISUAL ASSOCIATION MINING OF MULTIVARIATE DATA

A Dissertation Presented

by

Zhiyuan Zhang

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

May 2014

Stony Brook University

The Graduate School

Zhiyuan Zhang

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation.

Klaus Mueller – Dissertation Advisor
Professor, Computer Science Department

I.V. Ramakrishnan - Chairperson of Defense
Professor, Computer Science Department

Luis E. Ortiz
Assistant Professor, Computer Science Department

Kevin T. McDonnell
Associate Professor, Department of Mathematics and Computer Science, Dowling College

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Visual Association Mining of Multivariate Data

by

Zhiyuan Zhang

Doctor of Philosophy

in

Computer Science

Stony Brook University

2014

The rapid development of information technology produces vast amounts of data with numerous attributes. These multi-dimensional datasets offer tremendous opportunities for studying existing behavioral patterns and for predicting future developments. However, the high-dimensional space exceeds human comprehension. More sophisticated visualization techniques than the arsenal of standard plots are needed.

First, we introduce an interactive navigation technique to help the analysts explore within the multi-dimensional data spaces. We employ a network-based interface and pair it with a parallel coordinates plot. In the network interface, the dimensions form nodes that are connected by edges representing the strength of association between dimensions. The analysts can interactively manipulate a route in the network, which is captured by the parallel coordinates plot in the form of the dimension ordering. Then, we extend the navigation interface to interactive correlation and causation analysis for both numerical and categorical variables within a unified framework. We also build a landscape (map) out of the network, which shows the raw data within the network and helps analysts quickly learn relationships and trends of the data. We demonstrate it via several applications, such as helping statisticians with model discovery. Furthermore, we prove the viability of our framework in the context of real scientific problem—climate research, and show how it helps a team of scientists make important discoveries.

Finally, we introduce an interactive visual analytics interface designed for the healthcare informatics. It uses the Five-W's to establish a comprehensive multi-faceted assessment of the patient's history. The patient's multivariate data is visualized by associating each such W with a dedicated visual encoding that can represent and communicate it in effective ways.

To My Parents Xuesheng Zhang and Shumei Zhang,
my wife Xiaomeng Xu, and my son Aiden X. Zhang

CONTENTS

Contents v
List of Figures viii
Acknowledgements xii
Publications xiii
Chapter 1 Introduction 1
1.1 Problem Statement 1
1.2 Approach 2
Chapter 2 Background 5
2.1 Multivariate Data Visualization 6
2.1.1 Multidimensional Scaling (MDS) 6
2.1.2 Scatterplot Matrix 6
2.1.3 Parallel Coordinates Plot 7
2.1.4 Other Major Visualization Techniques 9
2.2 Visual Analytics 10
Chapter 3 Multivariate Data Visual Analytics Framework 12
3.1 Motivation 13
3.2 System Overview 14
3.3 Approach 15
3.3.1 Network Construction 16
3.3.2 Network Semantics 17
3.3.3 Network-Driven Dimension Ordering 18
3.3.4 Interactions with the Network 20
3.3.5 Focus + Context Browsing 21
3.3.6 Multi-Scale Zooming 23
3.3.7 Effect of Parallel Coordinate Display Interactions 24
3.4 Handling Categorical Variables 26
3.4.1 Theoretical Background 27
3.4.2 Dealing with Mixed Variable Pairs 28
3.4.3 Transforming the Categorical Variables 30
3.5 Integrating Data—the Subspace Scatterplot 31
3.5.1 Tessellating the Correlation Map 32
3.5.2 Generating the Subspace Scatterplots 32
3.5.3 Reading the Subspace Scatterplots 33
3.6 Using the Network for Story Telling with data 33
3.7 Correlation Analysis 35

3.7.1 Correlation Analysis: The University Dataset 35
3.7.2 Correlation Analysis: The Sales Campaign Dataset 36
3.7.3 Discussion 38
3.8 Subspace Scatterplot Based Analysis 38
3.8.1 Visualizing clusters and priorities 39
3.8.2 Finding appropriate subspace dimensionalities 39
3.9 Causation Analysis 40
3.9.1 Interactions 41
3.9.2 Causal Analysis 42
3.10 Evaluation 43
3.10.1 Dimension Ordering 43
3.10.2 Interface 45
Chapter 4 Geospatial data Analysis 47
4.1 Motivation 49
4.1.1 Domain Data 49
4.1.2 Domain requirements for the ISDAC dataset 51
4.2 System Design 51
4.2.1 Visualizing multivariate data (R1) 51
4.2.2 Summarizing different variables (R2) 52
4.2.3 Visualizing relationships (R3) 52
4.2.4 Supporting geo-spatial references (R4) 52
4.2.5 Supporting coordinated displays (R5) 53
4.3 Methods 54
4.3.1 Embedding the PHDs into Google Earth 54
4.3.2 Brushing in the Google Earth Display 55
4.3.3 Brushing in the PCP display 55
4.3.4 Addressing Exploratory Tasks 56
4.4 Use Cases and Results 56
4.4.1 Global seawater oxygen-18 database 56
4.4.2 ISDAC dataset 59
Chapter 5 Healthcare Analysis 63
5.1 Motivation 65
5.2 the Five W's Scheme 65
5.2.1 The Who and What 65
5.2.2 The Where 66
5.2.3 The When and Why 66
5.3 Encoding the Five W's 66
5.3.1 Hierarchical Radial Display 67
5.3.2 Sequential (Diagnostic Reasoning) Display 73
5.3.3 Implementation Details 77
5.4 Integration into Hospital Workflow 77
5.4.1 Diagnostic Assistant 77
5.4.2 Medical Coding Support 78
5.5 Usage Scenarios 79
5.5.1 Scenes From Daily Clinical Practice 79

5.5.2 Collaborative Analysis 81
5.6 Evaluation 84
5.6.1 Questions 84
5.6.2 Results 85
5.6.3 Coding support 87
5.7 Discussion 87
Chapter 6 Conclusion and Future Work 88
6.1 Future Work 88
Bibliography 91

LIST OF FIGURES

Fig. 2.1: Basic visualization charts. 5

Fig. 2.2: Scatterplot matrix of the car dataset. 7

Fig. 2.3: Parallel coordinates plots. (a) The five axes that represent the five dimensions. (b) PCP with one data point. (c) PCP with a few data points. 8

Fig. 2.4: Parallel coordinates plots. (a) PCP with all the data points. (b) Illustrative rendering of the PCP. The plot was color coded by the three pre-defined clusters. 8

Fig. 3.1: Storyboarding and storytelling with the Sales dataset. (a) Original dimension network display laid out by data correlations, along with automatically computed optimal route. (b) Linked parallel coordinate display [60] with axis order determined by the route in (a). (c) The user zooms into the network (blue rectangle in (a)) and manually specifies a route that seems to best capture the story – the strategic model of winning the most customers (see Section 3.6 for more detail), (d) Linked parallel coordinate display with updated axes ordering according to the route of (c). 12

Fig. 3.2: The network generation pipeline. 16

Fig. 3.3: Dimension layout in the Cars dataset via a mass-spring model. Higher color saturation on edges represents high correlation values. Green represents positive correlation, while red represents negative. (a) Layout with absolute correlation strength; (b) layout with positive correlation; (c) layout with negative correlation preference. 17

Fig. 3.4: Network-driven dimension ordering. (a) A tour of the Cars dataset. In the network display, the dimension ordering is given as a series of directed edges color-coded by correlation value. (b) Parallel Coordinate display. The dimension triples (*Horsepower*, *Cylinder*, and *Weight*) and (*Year*, *MPG*, and *Origin*) are put next to each other in the parallel coordinate display because they are strongly correlated with each other. These strong positive correlations can be seen on the network display also. (c) Correlation display colors the outlines of line bundles in terms of their correlation strength (less saturation maps to lower correlation). We can also discern negative correlations by the characteristic bow-tie shapes. 19

Fig. 3.5: User interaction and constraint imposition for the Cars dataset. (a) Network Display with Year being filtered out by multi-scale zooming. (b) The route is edited to avoid unrelated dimensions *Year* and *Origin*. (c) Routing with a cyclical constraint. Dimension Horsepower is visited more than once, which makes it adjacent to 4 dimensions (*MPG, Weight, Cylinders*, and *Acceleration*). The corresponding parallel coordinate display is shown in (d), where we can see that *Horsepower* has been duplicated. 20

Fig. 3.6: Focus+Context browsing. (a) Correlation network for the automobile dataset. (b) The edge correlation filter δe is set to 0.5 to remove low correlation edges. To get (c) and (d), first the user filters out *Height, Body, Door* and *Make* due to small accumulated correlations. (c) The vertex-browsing correlation network with focus only on *Price*. All edges that are incident on *Price* are highlighted. (d) The edge-browsing correlation network after selecting a group of variables (shown with thicker outlines). Only correlations (edges) within the group are shown to help the user to focus on the variables he/she is interested in. 22

Fig. 3.7: Multi-scale zooming merging criteria. (a) Variables *E* and *F* are positively correlated. As a result, both of them have the same correlation type with variable *D*. Then, when we merge *E* and *F*, the merged vertex retains the same type of correlation as before: *edge(D, EF)* vs. *edge(D, E)* and *edge(D, F)*. (b) Variables *B* and *C* are negatively correlated. Therefore, they have different correlations with variable *A*. If we merge *B* and *C*, the edge from *BC* to *A* will be inconsistent with those before the merging: *edge(A, BC)* vs. *edge(A, B)* and *edge(A, C)*. So we do not merge negatively correlated variables. 23

Fig. 3.8: Multi-scale zooming. Original correlation network is shown in (a). As one zooms out, (a), (b) and (c) show the result sequence of views of correlation network. From (c) we can see that variables *Weight, Length, Width, Price, Cylinder, and HP* have been merged into one (*Price*) because all are positively correlated. Although *MPG* is close to them, it has a negative correlation with the others, so it is not merged. But *MPG* and Drive Wheel have positive correlation, so they merge into one (*MPG*). The number of variables packed into the representative variable is given by the small number in the upper left corner of representative vertex. (d) and (e) show the corresponding PCP displays of (b) and (c), respectively. The representative dimensions are shown by double-line axes. 24

Fig. 3.9: The effect of dimension bracketing, using the global seawater oxygen-18 dataset. (a), (b) Parallel coordinate and network display of the original dataset with undefined values in dimension *depth*, *temp*, *salinity* and *d180*. (c), (d) After filtering out the undefined values and bracketing the dimensions. We can now see a strong positive correlation between salinity and *d18O* (green arrow in (d)), while in the original, unfiltered dataset with undefined values, *d18O* is incorrectly shown not to be strongly correlated with any other dimensions (green arrow in (b)). 25

Fig. 3.10: A synthetic data example that shows the necessity of categorical variable transformation in correlation analysis. (a) The original data—the set of crime rates for different spatial locations. (b) Randomly assign a number for each category (A→1, B→2, C→3, D→4). The corresponding correlation is -0.281. (c) Re-order the categories (A→3, B→2, C→1, D→4) while the spaces between categories are fixed (1 in this case). The corresponding correlation is 0.927. (d) Re-order and re-space, (A→3.5, B→2.085, C→1, D→3.48). The correlation is 0.97. The transformation reveals that there is a strong correlation between Location and Crime. 26

Fig. 3.11: Transformation results. (a): Correlation coefficients obtained by randomly assigning an integer value 1 through M to each of the M categories (Rand) and by using the spacing and ordering computed by our optimization (Opt): The correlations achieved by optimization are significantly higher, in many cases by an order of magnitude, for both datasets we tested: Auto and Car. (b), (c) and (d), (e) are two pairs of the parallel coordinate tiles that show the visual improvement after the transformation. 31

Fig. 3.12: Subspace scatterplot. 33

Fig. 3.13: Anatomy of a sales pipeline 34

Fig. 3.14: Correlation network for the University dataset. (a) Original correlation network. (c) After applying the edge correlation filter (setting δe=0.3) – the data divide into two fairly independent clusters – one (lower left) dealing with academic aspects, the other (upper right) with student life. 36

Fig. 3.15: Business strategizing with the sales campaign dataset (a) Aggregate correlation network for all three sales teams. (b) Parallel coordinate plot for the three sales teams (clusters), colored red, green, and blue. (c)-(e) Correlation networks for the red, green, and blue teams, respectively. All edges with correlation strength less than 0.2 have been filtered out to extract the main structure of the map. 37

Fig. 3.16: Scatterplot analysis: (a) default layout, (b) mesh after removing the edge between subspace 1 and 3 as well as 4 and 5. 39

Fig. 3.17: Difference between (a) the correlation map and (b) the partial correlation graph G_0. The two graphs have not been laid out by the mass-spring layout for better comparison. 42

Fig. 3.18: Visual causal analysis. (a) Mass-spring layout of the initial graph G0. (b) Graph G0 with lower partially correlated edges filtered out. (c) The potential causal graph after Step 2 and 3 of the TC-algorithm. 43

Fig. 3.19: Comparison of our automatic dimension ordering algorithm with other methods. (a) Original ordering. (b) The clutter-based ordering of Peng et al. [99] is unable to put the proper dimension order to show sub-clusters. (c) Ankerst's method [8] can capture the two subspaces, but the other un-related dimensions are split into two parts. Figures 8b and 8c are generated by xmdvTool [133]. (d) Ferdosi's subspace-based dimension ordering [38], which is able to capture the structure of the dataset (2 cluster subspace highlighted in red rectangle and 3 cluster subspace highlighted in blue rectangle). (e) Our method: the result is quite similar to (d). 44

Fig. 4.1: Interface and pie chart-histogram design (*PHD*). Panel 1 is the GE Object control panel which allows users to show/hide a GE object. Panel 2 is the bar-chart (or histogram) panel – here configured for *ISDAC* particle size. Its control panel allows the analyst to control various parameters, including the design of the *PHD* in the GE display. Panel 3 is the pie chart control panel. It contains the parameters for configuring the pie chart, such as what attributes will be displayed in the chart and which attribute is used to determine the size of the *PHD*. Users can also save/load the previous settings or export the current pie chart/histogram information into text files for further research. Panel 4 is the Google Earth display showing the three different *PHDs* styles we provide. When the user clicks on the *PHD*, the pie chart detail is shown nearby, and the corresponding size histogram is shown in the histogram panel. 47

Fig. 4.2: Capturing the ISDAC dataset. (a) The Single Particle Mass Spectrometer (SPLAT II) operated by the collaborating scientist in-flight in the Arctic aboard a Convair-580 research aircraft. (b) Various sensor probes mounted on the aircraft wing. The aircraft flew various missions over Alaska to measure concentrations, size distributions, shape, density, and compositions of millions of particles in clear atmosphere to establish a large and highly resolved data set of Arctic aerosol particles. Other environmental variables, such as cloud density, pressure, and density were also sampled (c) Overview of the flight path and (d) profile as seen from the side, both captured with Google Earth. 50

Fig. 4.3: Dual-domain analytics. (a) The analyst first uses the PCP brushing handles to select the normal ocean data points (salinity from 32 to 40). (b) The GE display responds by showing only these remaining data points. (c) Next the analyst uses mouse clicks to outline some interesting regions in the GE display (Mediterranean shown in green and Gulf of St Lawrence shown in red). (d) The points inside the selection polygon appear highlighted in the PCP display. (e) Correlation-enhanced PCP display. 53

Fig. 4.4: Flowchart showing how to dynamically render an object into Google Earth. Here the pie chart is used as an example. . 54

Fig. 4.5: Interactions in the PCP display. (a) PCP and (b) correlation-enhanced PCP display of the original dataset with undefined values in dimensions *depth, temp, salinity* and $\delta^{18}O$. (c) and (d) The same set of displays after filtering out the undefined values and zooming into the dimensions. We now have a clearer view of the remaining data. For example, we can see a strong positive correlation between *salinity* and $\delta^{18}O$ (green square in (c) and (d)) after the dimension zoom-in, while in the original dataset with undefined values, $\delta^{18}O$ is falsely negative correlated with *salinity* (red square in (b)). (e) Correlation display after axis depth inversion. Sea *Depth* and *Temperature* now have a positive relationship (blue square in (e)), which is consistent with our knowledge. 57

Fig. 4.6: *Salinity* is nearly 0 psu while the $\delta^{18}O$ has widely differing values due to river inflow. (a) Brushing in PCP to select data that have near-zero salinity. These areas are: (b) Obskaya Gulf (estuary of Ob River), Yenisey (estuary of Mal. Taz River) and White Sea in Russia. (c) St. Lawrence River area in Canada. 58

Fig. 4.7: Deep seawater (*depth*>2000) analysis. Both displays show that at deep sea the variation of $\delta^{18}O$ is larger than salinity. 58

Fig. 4.8: An overview of particle compositions and size changes along the flight. The flight track is marked as a red line and each of the one-minute-spaced data points is superimposed as a grey ellipse. The polygon selection tool is used to outline several interesting areas (indicated by yellow polygons and labeled in red) and the corresponding *PHDs* are drawn nearby. *PHDs* are sized by the variable *NSplat* to allow an assessment of compression. The colors for the pie charts are assigned by the scientists via a color map popup widget, applying their domain standards. This overview visualization shows significant spatial variability in particle composition and size, which is consistent with previous reports that show a highly stratified atmosphere. 60

Fig. 4.9: Cloud data analysis. Here the scientists first used the PCP display to filter data points with clouds (*Nd*>20). They then applied a second filter to select only data points in which cloud droplets are sampled (*CVI* =1, green points on the flight track) to make the upper *PHD*. They then changed the filter to select data points where only the particles between the cloud droplets are sampled (CVI=0, blue track points) and obtain the lower *PHD*. The fact that these two *PHDs* are virtually identical is a significant discovery in climate research. This has never been done before and changes much of what has been assumed in the field. 61

Fig. 4.10: Changes in particle composition as a function of altitude. We zoom into the flight's spiral ascent, where the aircraft climbed from a few hundred meters to an altitude of about 7000 meters. The pie charts clearly illustrate that particle compositions change significantly with altitude and that the changes are not monotonic. Here we use the first *PHD* rendering method (G1) in Secion 4.3 because it better shows the alleviation. 62

Fig. 5.1: The two coordinated displays of our system. (a) The radial (patient overview) display with integrated body map, along with the user interface. (b) The corresponding sequential (diagnostic reasoning) display using the same color coding. The user interface is identical to the radial display but removed here to save space. 63

Fig. 5.2: Evolution of the medical record. (a) Paper-based. (b) Electronic. Images were shrunk to hide the patient's information. 64

Fig. 5.3: Node design. Shade of red encodes severity. This node tells us that the patient has a relatively severe disease in the nervous and sensory organs. There have been a total of three doctor visits related to these diseases. The children layer gives more detail on the specific kind of disease within this broad category. 68

Fig. 5.4: Color Composition. Node A contains three doctor visits; Node B contains only one but very recent doctor visit. (a) Using the MAX operator. (b) Using color composition. The coloring of (a) suggests that node A has the highest priority. But A's color is determined by a severe disease which however occurred in the patient's first doctor visit (a long time ago). That disease might not be as important compared to some less severe diseases that occurred in the patient's last visit (just a few days ago). So, if more recent occurrences are the focus, this disease will be better highlighted using color composition (shown in (b)) which takes time into account. This makes node B more apparent. 70

Fig. 5.5: Node collapse strategy. (a) Original edges e1 and e2 link the incident nodes; (b) Node [222-224] and [123] collapse. Edges linked to their children now link to themselves. (c) Only the first level nodes are shown, then e1 and e2 merge into one single edge. The opacity of the edge is computed by compositing those of e1 and e2. 71

Fig. 5.6: Different layouts for the sunburst display with two alternative node labeling schemes. (a) The hierarchy-centric layout. The nodes are labeled with the ICD9 codes but labeling with the corresponding medical terms is also possible. (b) The patient-centric layout. Here the nodes are labeled with the corresponding (abbreviated) medical terms as available in the ICD9 description. 71

Fig. 5.7: Some features of the sequential display: information text window, post-it note, and red back-edge. A back-edge from a treatment to an incident can denote a referral, while a back-edge from a treatment to a symptom can denote a side-effect as is the case here – the drug *Valacyclovir* prescribed at a previous visit (Dec. 3) causes the current visit's (Dec. 8) symptoms of nausea and vomiting. 74

Fig. 5.8: Severity and uncertainty rating bar variations for sequential display nodes. The location of the vertical black line below the node indicates whether the data item is a two-sided rating (black line in the middle) or a one-sided rating (black line on the left). The length of the bar encodes the value and its saturation encodes the uncertainty of that value. The text inside the boxes above explains the semantics that our system uses in more detail. 75

Fig. 5.9: Sequential (diagnostic reasoning) display. (i) Node-collapse to focus on the most recent visits – a total of more than 100 incidences is shown. (ii) In the browsing highlight mode, the doctor has selected one of the diagnoses which highlights all related nodes and branches but fades out the others. The same type of highlighting also occurs when filtering. 76

Fig. 5.10: Diagnosing a case of *Hypokalemia* with the sequential display. The cause is a recently prescribed medication: *Lasix*. 80

Fig. 5.11: Radial displays for two patients reporting to the ER with back pain. The relevant area is the lower left quadrant labeled Mus. Conn. From the time histograms we see that patient A had no incidence of back pain before, while patient B is a frequent sufferer. 81

Fig. 5.12: Complex medical case involving four different doctors in a collaborative diagnosis task. Top: sequential display after the fourth visit. Bottom: the emerging radial displays labeled by visit number. 83

ACKNOWLEDGEMENTS

The six-year stay at Stony Brook University is a precious experience in my life. I would like to begin by expressing my deepest gratitude to my advisor, Professor Klaus Mueller, for his continual and thoughtful guidance, advice and support. He led me to the field of information visualization and visual analytics. He has been inspiring, encouraging and supportive in every way during my study and life. I am proud to join his rich legacy of Ph.D. graduates. I am thankful to my committee members, I.V. Ramakrishnan, Luis Ortiz, and Kevin McDonnell for their valuable critique, suggestion, and assistance. Without their support, this dissertation would not be possible. I am also grateful for Distinguished Professor Arie Kaufman, for his guidance and support during my first year of my Ph.D. study.

I would like to thank my mentors, managers and colleagues during my internship, Zoe Abrams, Itamar Rosenn, Siyang Chen, Paul Jones, Eric Zilli from Facebook, and David Gotz, Adam Perer, Jianying Hu, Jimeng Sun, and Fei Wang from IBM Research, for their support of my off-campus research and life. It was such a pleasure working with them, and I gained valuable knowledge and experience that are essential to my Ph.D. study.

I would also like to thank all my colleagues, friends and staff members at the Computer Science Department who have helped me in these years. I want to thank Stella Mannino and Cynthia Scalzo for their generous help. I want to thank all my research colleagues, especially those in the Visual Analytics and Imaging Lab, for their delightful discussions and collaborations: Wei Xu, Ziyi Zheng, Bing Wang, Hyunjung Lee, Nafees Ahmed, Sungsoo Ha, Eric Papenhausen, Puripant Ruchikachorn, Quoc Duy Vo, Shenghui Cheng, Jisung Kim, Lei Wang, Song Feng, and Tianyun Ling.

Last but not least, I want to thank my wife, Xiaomeng, my son, Aiden, and my parents. I cannot write in words how much support and encouragement I have received from them through these years and I am truly grateful to have them by my side.

PUBLICATIONS

Z. Zhang, K. McDonnell, E. Zadok, K. Mueller. Visual Correlation Analysis of Numerical and Nominal Data on the Correlation Map. (Accepted, to appear), *IEEE Transactions on Visualization and Computer Graphics.*

Z. Zhang, D. Gotz, A. Perer. Iterative Cohort Analysis and Exploration. *Information Visualization.* 1473871614526077. March, 2014.

Z. Zhang, B. Wang, F. Ahmed, I.V. Ramakrishnan, R. Zhao, A. Viccellio, K. Mueller. The Five W's for Information Visualization with Application to Health Informatics and Electronic Medical Records. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1895-1910. 2013.

S. Cheng, Z. Jiang, Z. Zhang and K. Mueller. A Visual Analytics System for Stock Data. *IEEE Visualization Conference (poster)*, Atlanta, GA, Oct., 2013.

Z. Zhang, X. Tong, K. McDonnell, A. Zelenyuk, D. Imre, K. Mueller. An Interactive Visual Analytics Framework for Multi-Field Data in a Geo-Spatial Context. *Special Issue of Tsinghua Science and Technology on Visualization and Computer Graphics.* 18(2):111-124. April, 2013.

A. Zelenyuk, D. Imre, Z. Zhang, J. H. Lee, K. Mueller, K. McDonnell. An Interactive Visual Analytics Framework for Multidimensional Data in a Geo-Spatial Context. *American Association for Aerosol Research (AAAR) 32nd Annual Conference*. Portland, OR, Sept. 2013.

L. Kühne, J. Giesen, Z. Zhang, S. Ha, K. Mueller. Data-Driven Approach to Hue-Preserving Color-Blending. *IEEE Transactions on Visualization and Computer Graphics.* 18(12):2122-2131. 2012.

Z. Zhang, D. Gotz, A. Perer. Interactive Visual Patient Cohort Analysis. *IEEE Workshop on Visual Analytics in Health Care.* Seattle, WA. Oct. 2012.

Z. Zhang, K. McDonnell, K. Mueller. A Network-Based Interface for the Exploration of High-Dimensional Data Spaces. *IEEE Pacific Visualization 2012 (Back cover image)*. Seoul, Korea, pp. 17-24. March 2012.

Z. Zhang, F. Ahmed, A. Mittal, I.V. Ramakrishnan, R. Zhao, A. Viccellio, K. Mueller. AnamneVis: A Framework for the Visualization of Patient History and Medical Diagnostics Chain. *IEEE Workshop on Visual Analytics in Health Care,* Providence, RI. Oct. 2011.

Z. Zhang, A. Mittal, S. Garg, A.Dimitriyadi, I.V. Ramakrishnan, R. Zhao, A. Viccellio, K. Mueller. A Visual Analytics Framework for Emergency Room Clinical Encounters. *IEEE Workshop on Visual Analytics in Health Care.* Salt Lake City, UT. Oct. 2010.

S. Ha, Z. Zhang, S. Matej, K. Mueller. Efficiently GPU-Accelerating Long Kernel Convolutions in 3-D DIRECT TOF PET Reconstruction via a Kernel Decomposition Scheme. *IEEE Medical Imaging Conference.* Knoxville, TN. Oct. 2010.

Chapter 1

INTRODUCTION

1.1 PROBLEM STATEMENT

In the era of information explosion, the ultimate goal for data analysts and researchers is to understand the data collected and make useful decisions from them. More often than not, the data that are collected in scientific researches will have multiple variables (also known as dimensions, attributes). For example, a data set of automobile can have variables such as production year, assembly country, weight, length, width, height, expected miles per gallon, number of cylinders, and so on; a business sales data may contain variables such as number of leads, number of opportunities, revenue, cost, expected return of investment, and so on. These data with multiple variables can be regarded as multivariate (or multi-dimensional) data. Nowadays, multivariate data have become ubiquitous in a wide range of domains, such as science, finance, business, demographics, biology, healthcare, and the like. Unfortunately, the multi-dimensional space exceeds human comprehension.

The navigation of multi-dimensional data spaces remains challenging, making multivariate data exploration difficult. The multivariate datasets offer tremendous opportunities for studying behavioral patterns and also for predicting future developments. The most valuable insight often comes from intricate inter-relationships among data attributes, and extracting these relationships requires skills in multi-dimensional data understanding. For example, in psychology research, scientists try to find relationships between intelligence, aptitude and social behavior. In finance and economics, to maximize profit, economists look for the group of variables that are mostly related to profit. Finally, in the social and natural sciences, researchers seek to understand and explain the nature of relations between phenomena. To make progress in this wide gamut of areas, analysts require effective, sensitive, and intuitive tools to uncover these relationships.

Correlation analysis is one such tool. It looks for relationships between variables and can show whether pairs of variables are related and how strongly. Correlation analysis has become increasingly popular in psychology, education, finance, marketing, and climatology, just to name a few. Correlations, however, are difficult to interpret, manage, and survey once the number of variables becomes even moderately large. Given D variables, there are $O(D^2)$ correlation pairs,

which makes complex relationships difficult to recognize from columns of numbers alone. Hence, there is a clear need for an effective visual interface that allows analysts to (1) quickly get an overview of the overall correlation relationships in the data, and (2) easily manipulate the data to reveal hidden relationships via different modes of interactions, such as filtering, selection, bracketing, and clustering.

Furthermore, realistic datasets often contain a mix of numerical and categorical variables. Although there are well-defined statistical techniques to handle categorical and numerical variables in isolation, mixtures of these have received less attention. This has a great influence on the sensible computation of correlation factors. How to deal with numerical and categorical variables within a unified framework has been not only a research topic of great interest, but also one of the great challenges in the visual analytics domain.

Finally, causation analysis looks beyond correlation relationship. Causation analysis is an analytical process that tries to answers the questions of causes, reasons, effects, results and consequences. Often it helps analysts make better judgments or predictions by knowing the causes and effects. Here we are focusing on the problem of causation analysis from observational data. Pearl [99] proposed to use graphical model for causal reasoning. The basic idea is to use a *Directed Acyclic Graph* (DAG) to represent the causal relationships, or at least the probabilistic dependencies, between them. In the causal graph, each variable is represented as node. If variable *A* is a direct cause of *B*, then there is an edge going from *A* to *B*. In this case, we also say that *A* is one of *B*'s parents, and *B* one of *A*'s children. If there is a (causal) path from *A* to *B*, then *A* is an ancestor of *B*, and *B* is a descendant of *A*. However, how to build such a causal graph from multivariate data has posed a great challenge for the researchers due to the curse of dimensionality. Moreover, like many other analytical and statistical algorithms, traditional causal model identification algorithms lack of user interactions. The analysts have little influence on the automated analytical process. However, the analysts' expertise should play an important role in the whole process.

1.2 APPROACH

It is well known that graphical displays and visualization can show data patterns more clearly than can plain text and numbers. However, because there is no simple mapping of the multiple dimensions onto a two-dimensional screen, more sophisticated visualization techniques than the arsenal of standard plots are needed.

To address these needs, we first describe a network-based interface coupled with an interactive parallel coordinates plot to assist the analysts in exploring the multi-dimensional data spaces. In the network display (which we also call the Correlation Network), vertices correspond to variables and are rendered as filled circles. A vertex's size is initially determined by the corresponding variable's coefficient of variance, but users may interactively resize vertices as desired. The vertices are laid out via a mass-spring model in which spring rest length is a

function of the pairwise Pearson's correlation coefficient. As such, the layout can help users to gain a quick understanding of the correlations existing in the data by comparing the distances of the nodes representing the variables. The edge is weighted by the correlation between the two dimensions linked by the edge. The correlation is encoded by color. Green encodes a correlation value of +1 while red encodes correlation value of -1, and 50% gray is used to encode 0. Linear interpolation computes the colors in between. On the other hand, the placement of the vertices and the edges drawn between them then provide the guidance for navigation and exploration. The network implements the dimension ordering interface as an interactive route planner, operating in a data-informed network where vertices represent dimensions and edges connecting two vertices represent promising dimension pairings. Users can change the route via mouse clicks, and the interface then computes a new optimized route that includes the changed route portion. Other capabilities include multi-resolution navigation and zooming of the network display. Conversely, parallel coordinate plots allow bushing and filtering to focus on certain sub-patterns of interest which can be used to reconfigure the correlation network. It is these mutual benefits that suggest a tightly integrated framework of the two constituents: correlation network and parallel coordinates.

To deal with the mixture of numerical and categorical variables within one unified framework, we devise a scheme that transforms the categorical variable to numerical one. This not only avoids the problems with binning that was introduced by transforming numerical variable into categorical one, but also enables the use of Pearson's correlation exclusively. It also allows us to find, for a given categorical-numerical variable pair, the spacing and order of the categorical variable's levels that maximize correlation. In order to maximize the correlation between the categorical-numerical variable pair, the task we are faced with is to determine the spacing and ordering of the levels of a categorical variable with respect to a numerical variable. This can be accomplished by solving a regression model in which the categorical variable is the independent variable—one dummy variable per level less baseline—and the numerical variable is the dependent variable. The coefficients returned by the least squares optimization then determine the desired order and spacing of the corresponding categorical levels.

To extend our correlation network for causation analysis, we modify our correlation network by using a partial correlation based method for causal model discovery. As we can see that our correlation network and Pearl's causal graph [99] share the same underlying concept: vertices correspond to variables and edges encode relationships. Although in many cases, the more robust the correlations, the more likely they are to imply causation, using the correlation map directly for causal reasoning could be problematic. Hence, we modify our correlation network based on the idea that was first presented by Pellet and Elisseeff [101]. The basic algorithm works as follows:

(1) For each pair (*X, Y*), add an edge $X-Y$ if the partial correlation $\rho_{XY \cdot V\backslash\{X, Y\}}$ does not vanish. We obtain the moral graph of G_0, i.e., an undirected copy of G_0 where all parents of the colliders are pairwise linked;

(2) Remove spurious links between parents of colliders introduced in Step (1) and identify V-structures;

(3) Propagate constraints to obtain maximally oriented graph.

We apply the mass-spring model layout to the network computed from the partial correlations in Step (1). The layout would place strongly partial correlated variables close to each other. In other words, highly conditionally dependent variables would be put close to each other. Consequently, these nearby variables would serve as a good starting point for further investigation of the underlying causal effects. We also add various user interactions into the framework to allow analysts to interactively modify the constraints and dependencies based on their expertise in the causal model discovery process.

The remainder of the dissertation is organized as follows. Chapter 2 gives a brief introduction of the basic concepts and background. Chapter 3 introduces a new interactive navigation technique to help analysts explore within the multi-dimensional spaces. Then we extend the navigation technique and utilize the interactive framework for correlation analysis and causation analysis. Chapter 4 applies the framework to real scientific problems – climate research, and demonstrates how the framework could help scientists to solve their climate research problems. Chapter 5 introduces another interactive visual analytics framework that is designed specifically for the healthcare application. It uses the Five W's (*when, who, what, where, and why*) as a means to establish a comprehensive multi-faceted assessment of the patient and his/her history. Chapter 6 gives conclusion and directions of the future works.

Chapter 2

BACKGROUND

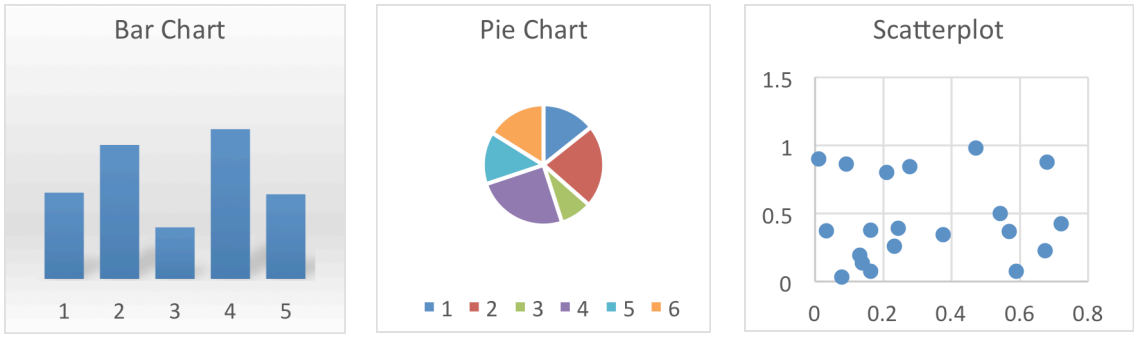


Fig. 2.1: Basic visualization charts.

Visualization is a technique that uses graphical representations, such as images, diagrams, or animations to help explore data and information, gain insight into information space, and to reinforce cognition, hypothesis building and reasoning. The two majorities of visualization are scientific visualization and information visualization.

Scientific visualization focuses mainly on data from simulations or experiments, with an implicit or explicit geometric structure. It aims to help explore, analyze and understand the data. Traditional areas of scientific visualization are volume visualization, flow simulation, medical imaging, etc.

Information visualization focuses mainly on large amount of multivariate data. It aims to use visual representation to allow people to directly interact with the data, and to help people get insight into the data, draw conclusions, and make decisions. Information visualization serves as a good tool to do data mining and knowledge discovery visually. Compared to the raw data, numbers, or texts, this is more intuitive and easily understood for the user. The Mantra for information visualization [121] is “Overview first, zoom and filter, then details-on-demand”. It means that the visual representation firstly should provide the users with a clear overview of the whole dataset; then the users can interactively select the interesting portion of the data and zoom into those data points to see more detailed information. This report will focus on the techniques of information visualization.

Visualizations have been widely used to represent the data and to help uses better understand the data. Some very basic visualization techniques that are familiar to the public are bar chart, line chart, pie chart, scatterplot. Fig. 2.1 shows some of the examples. All these popular techniques are designed to visualize data with only one or two variables (dimensions) by intuitively mapping the variables to the two dimensional display. However, as the number of the dimensionality increases, people have more and more difficulties to understand the data. First, there is no simple mapping of the multiple dimensions to a two-dimensional display. Second, the high dimensionality exceeds human comprehension. Finally, the high number of dimensions makes it extremely difficult to spot the clusters, outliers, and the relationships. As a result, more sophisticated visualization techniques than the arsenal of standard plots are needed.

2.1 MULTIVARIATE DATA VISUALIZATION

Following describes some major techniques to visualize multivariate data.

2.1.1 MULTIDIMENSIONAL SCALING (MDS)

Multi-dimensional scaling (MDS) [77][130] is series of methods that try to embed the data into a lower dimensional space (1D-3D), which is essentially a dimension reduction technique. MDS is widely used since it groups similar data items close together, given some distortion, and so provides an overview of the pattern of proximities (i.e., similarities or distances) among a set of objects.

For example, given the N-dimensional dataset A with M data points, we can construct a $M \times M$ dissimilarity matrix D, where each element $d_{i,j}$ represents the dissimilarity information of a_i and a_j based on some specific similarity function.

The goal of MDS is to map the N-D data points a_i, ($i = 1; \ldots; M$) to low dimensional (e.g. 2D) data points x_i, ($i=1; \ldots; M$) in such a way that the given dissimilarities $d_{i,j}$ are well-approximated by the distances between x_i and x_j. That is, the smaller distance that two data points have, the less dissimilar (or in other word more similar) these two points are; and vice versa. Then by looking at the 2D mapping of the high-dimensional dataset, we can quickly understand the proximities of the dataset. The main drawback of MDS is that the dimension information is lost after the mapping and it is difficult to project back from the low dimensional space to the high dimensional space.

2.1.2 SCATTERPLOT MATRIX

Scatterplot matrix [53] provides a simple, familiar, and clear view of the data distributions. The scatterplot matrix is built by first arranging dimensions vertically and horizontally, and then plotting basic scatterplot for each pair of dimensions. In other words, if the dataset has N variables, the corresponding scatterplot matrix will have N rows and N columns and the i^{th} row and j^{th} column of this matrix is a basic scatterplot of the i^{th} variable v_i versus the j^{th} variable v_j. Fig. 2.2 shows an example of scatterplot matrix of the car dataset [156].

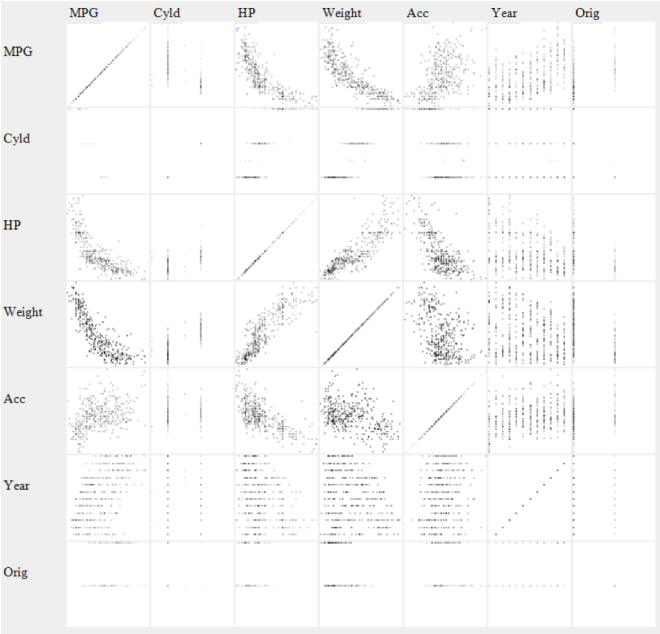


Fig. 2.2: Scatterplot matrix of the car dataset.

There are some techniques to simplify the scatterplot matrix. First, for the diagonal plot, since the two dimensions come from the same variable, it will be simply a 45-degree line. Although the density of the line can help to show the univariate distribution of the variable, it is not as good as a simple histogram. Some other alternatives are also popular. Some prefer to use the diagonal to print the variable label; while others would like to simply leave the diagonal blank to make the matrix more concise. Second, since the plot of v_i versus v_j is equivalent to the one of v_j versus v_i with the axes reversed, we can simply omit the plots below the diagonal to save more space.

The scatterplot matrix provides a familiar way to show the relationships of all the dimensions in the dataset simultaneously. However, due to their distributed 2D tiled layout, it can be difficult to discern relationships that extend across and involve more than two variables.

2.1.3 *Parallel Coordinates Plot*

Finally, parallel coordinates plot (PCP) [60] shows the raw high-dimensional data points as polylines spanning across a set of parallel vertical axes, each representing one dimension. PCP unifies the advantages of MDS and scatterplot matrix: it can convey all data dimensions in a single display, as well as preserves the original data dimensions. The visual analytics framework in this report was developed based on the PCP, so following we shall give some details about the PCP.

In the following example, we use a simplified version of the car dataset [156] that has only 5 variables we wish to explore: MPG (ranging 9-47), Cylinders (3-8), HorsePower (46-230), Weight (1613-5140), and Acceleration time (8-25).

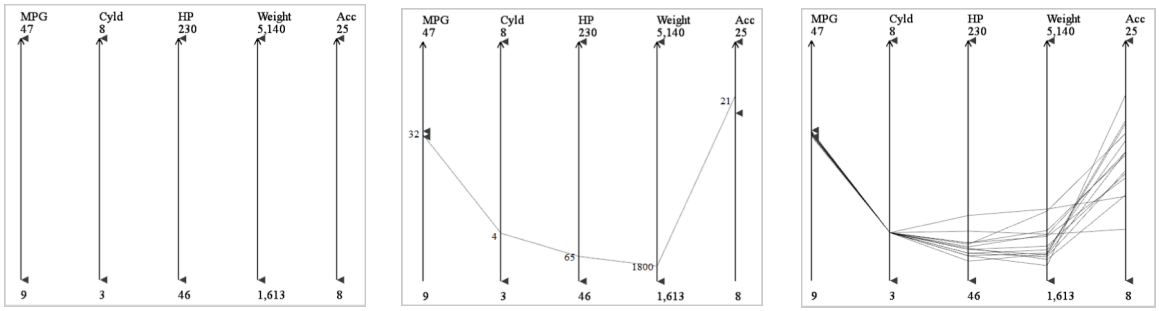


Fig. 2.3: Parallel coordinates plots. (a) The five axes that represent the five dimensions. (b) PCP with one data point. (c) PCP with a few data points.

To create a PCP, we first vertically lay out one axis for each variable in the dataset (Fig. 2.3(a)). Each data point in a dataset has a particular value for each of the 5 variables. For example, given a car with MPG = 32, Cylinders = 4, HorsePower = 65, Weight = 1800 and Acceleration = 21, the system determines the intersection points on each of the 5 axes based on the corresponding value locations. Then we can connects those intersections with one polyline to create a multi-axis representation of a single data point, as shown in Fig. 2.3(b). Fig. 2.3(c) shows the result after adding a few more data points. Now we can start seeing the trend in the data – cars with relative high MPG will have relative low number of cylinders, low horsepower, low weight, and relative high acceleration time.

Fig. 2.4(a) shows the result after adding all of the data points in the dataset. Though we can still see the trend of polylines, they become hard to follow. Following are the two major issues of the PCP:

(1) As we have more and more data points, the cluttered display could be one big problem for readability of the PCP.

(2) The information that can be visually extracted from the PCP is highly dependent on the ordering of the dimensions.

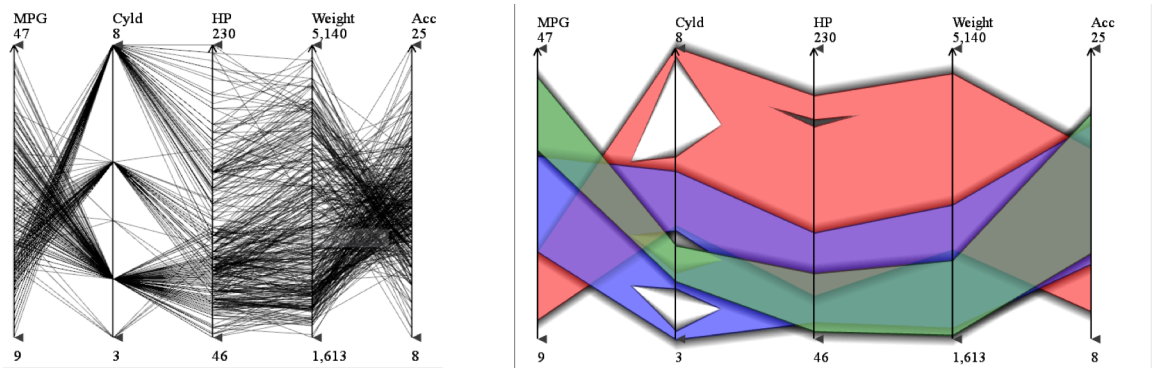


Fig. 2.4: Parallel coordinates plots. (a) PCP with all the data points. (b) Illustrative rendering of the PCP. The plot was color coded by the three pre-defined clusters.

To alleviate the clutter problem, a number of techniques have been proposed: using free-form curves in place of the polylines [46][87][128][146][150], performing clutter reduction via density analysis [99] or clustering [49][81], or using illustrative rendering techniques to simplify the display with a more aesthetic appearance [87]. Fig. 2.4(b) shows an illustrative rendering result of the parallel coordinate plot. When the number of lines increases, it is really difficult to follow each line. Most perceptions that the viewers can observe are just the trend of all the lines. So compared to the original raw line rendering, this illustrative rendering did not lose much information, but with a better display of the trend and the more aesthetic looking.

Dimension ordering in parallel coordinates has been studied for quite some time. Many methods do this fully automatically. Inselberg and Avidan [61] devise a classifier for both dimension selection and ordering. Ankerst et al. [7] define a similarity metric (either partial or global) that compares two dimensions by the RMS distance of all data points, and then optimize an ordering by ways of an approximate traveling salesman (TSP) solver – here an ant colony optimization. Conversely, Tatu et al. [127] propose a similarity-based function based on Hough Space transforms. Johansson and Johansson [64] define a weighted metric that rates dimensions by their importance with regards to correlation, outlier, and subspace cluster importance and use this rating to select relevant parallel coordinate dimensions. Similar to Artero et al. [11] they then find correlation of dimension pairings to optimize this ordering. Dasgupta and Kosara [32] devise several metrics based on the appearance of the polyline display to find good dimension orderings. Peng at al. [99] determine an ordering that minimizes clutter and outliers between adjacent dimensions. Finally, Ferdosi and Roerdink [38] order the dimensions according to high-dimensional structures identified by sub-space clustering. Fua et al. and Yang et al. [43][141][143] use hierarchical dimension filtering in conjunction with dimension clustering via Ankerst's metric. Then they allow users to interactively navigate the hierarchy to produce a preferred ordering. Elmquist et al. [35] apply Ankerst's metric to arrange scatterplot matrix tiles into a 2D gridded layout and then visualize the transition across tiles with 3D plots. As such, the grid functions as a map, where the routes travel along the horizontal and vertical grid lines.

2.1.4 OTHER MAJOR VISUALIZATION TECHNIQUES

Similar to parallel coordinates plot, Kandogan [66] proposed Star coordinates plot, which arranges coordinates on a circle sharing the same origin at the center. Star coordinates plot uses simply points to represent data, treating each dimension uniformly at the cost of coarse representation. Ambiguities can arise when two points with different attributes are coincidently located at the same position. Klippel [73] proved that color enhanced star plot glyphs have positive effects on the processing speed and that they reduce the influence of salient shape characteristics. Elmqvist et al. [34] also use the concept of star plot to build an interactive tool that allow users to create advanced visual queries by selecting and filtering into the multidimensional data.

There are also some other techniques that also try to address the representation of multi-dimensional data in many different ways. Yi et al. [144] proposed Dust&Magnet, which used a

magnet metaphor to represent the multivariate dataset. Mosaic plot can also be used to visualize multi-dimensional dataset by starting with a big rectangle, then keeping splitting for each additional dimension [41][90][42][129]. Dense pixel display [69] is another technique to show high dimensional data. The basic idea is to map each dimension value to a colored pixel and group the pixels belonging to each dimension into adjacent areas. Since dense pixel displays use one pixel per data value, the techniques allow the visualization of the huge amount of data possible on current displays. The question is how to arrange the pixels on the screen. By arranging the pixels in an appropriate way, the resulting visualization provides detailed information on local correlations, dependencies, and hot spots. Well-known examples are the recursive pattern technique [70] and the circle segments technique [8]. Another type of multi-dimensional techniques is to map the attribute values of the data onto the features of an icon. Icons can be arbitrarily defined: such as faces [24], needle icons **Error! Reference source not found.**, stick figure icons [103], color icons [71][80], and TileBars [54]. Then by looking at the features of the icon, users can compare and get a sense of what the attribute values are.

Finally, the various paradigms can also be integrated. Schmid and Hinterberger [119] combine scatterplot matrices, parallel coordinates, permutation matrices, and curve display together. Wong et al. [139] combine and link parallel coordinates with scatterplots matrices, and Yuan et al. [146] integrate parallel coordinates with scatterplots, using MDS to convert multiple axes into a single subplot.

2.2 VISUAL ANALYTICS

To help users have a better understanding of the data, various data analysis techniques are proposed to analyze the data, build models for the data, and extract patterns from the data. However, most of the techniques work like a black box, the users have little control over the analytical process. For example, users cannot interactively customize the inputs of the algorithm, or interactively refine some parameters of the algorithms based on the observations or algorithm outcomes. To bridge the gap, visual analytics integrates visualization, data mining, and statistics

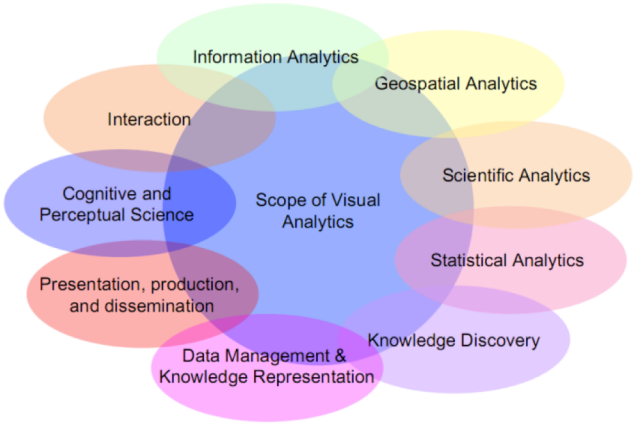


Figure 2.5: Scope of Visual Analytics.

with interactive visual interfaces that allows the users to apply their domain knowledge in the traditional automatic analytical process. Keim give a more detailed definition – visual analytics is an iterative process that involves information gathering, data preprocessing, knowledge representation, interaction and decision making [67][68]. It combines the computation capabilities of machines with the perceptual and reasoning strengths of humans. On the one hand, methods from knowledge discovery in databases, statistics and mathematics are the driving force on the automatic analysis side, while on the other hand human capabilities to perceive, relate and conclude turn visual analytics into a very promising field of research. The scope of visual analytics is shown as shown Fig. 2.5.

The advantages of visual analytics over traditional data mining techniques are:

- Visual analytics can easily deal with highly inhomogeneous and noisy data
- Visual analytics is intuitive and requires no understanding of complex mathematical or statistical algorithms or parameters.

Visual analytics usually follows the information visualization mantra three-step process: overview first, zoom and filter, and then details-on-demand [121]. First the system gives the user a visual overview of the whole dataset. The visual representation of the information reduces complex cognitive work needed to perform certain tasks and helps the user gain insight of the data quickly and accurately. All the information visualization techniques that were talked about in the previous section can be used. After identifying some interesting patterns, the system should allow user to interactively filter down and focus on one or more subsets of the data. Note that visualization techniques not only provide the basic visualization techniques for all the three steps but also bridge the gaps between the steps.

Interaction is an essential part of visual analytics. Instead of the low-level interaction techniques, Yi et al. [145] propose seven general categories of interaction techniques that are organized around a user's intent while interacting with a system:

1) Select: mark something as interesting
2) Explore: show me something else
3) Reconfigure: show me a different arrangement
4) Encode: show me a different representation
5) Abstract/Elaborate: show me more or less detail
6) Filter: show me something conditionally
7) Connect: show me related items

As discussed above, one primary technical barrier is that the high dimensionality (high number of variables) exceeds human comprehension, and so, effective tools are needed to boost these capabilities. The following of the report is dedicated to address this problem.

Chapter 3

MULTIVARIATE DATA VISUAL ANALYTICS FRAMEWORK

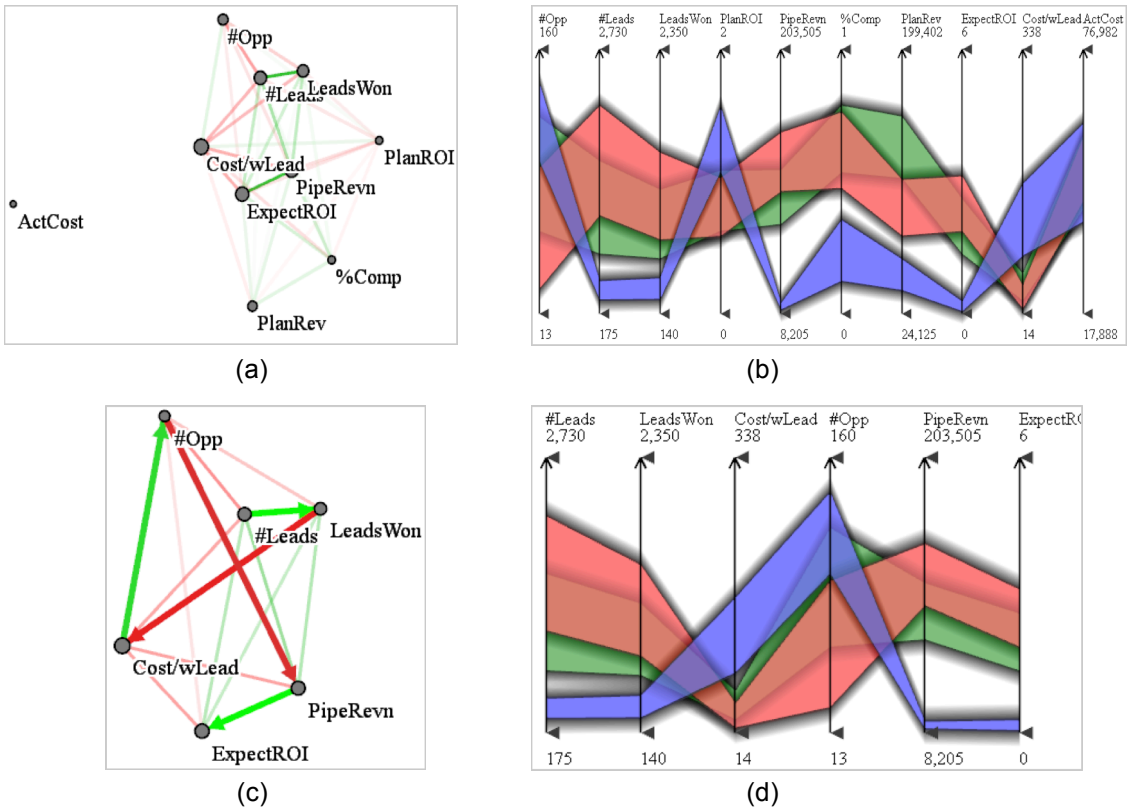


Fig. 3.1: Storyboarding and storytelling with the Sales dataset. (a) Original dimension network display laid out by data correlations, along with automatically computed optimal route. (b) Linked parallel coordinate display [60] with axis order determined by the route in (a). (c) The user zooms into the network (blue rectangle in (a)) and manually specifies a route that seems to best capture the story – the strategic model of winning the most customers (see Section 3.6 for more detail), (d) Linked parallel coordinate display with updated axes ordering according to the route of (c).

The navigation of multi-dimensional data spaces remains challenging, making multivariate data exploration and analysis difficult. To be effective and appealing for the mainstream application, navigation should use paradigms and metaphors that users are already familiar with. One such intuitive navigation paradigm is interactive route planning on a connected network. We

have employed such an interface and have paired it with a prominent multivariate visualization paradigm showing the N-D data in undistorted raw form: parallel coordinates plot. In our network interface, the dimensions form nodes that are connected by a network of edges representing the strength of association between dimensions. A user then interactively specifies nodes/edges to visit, and the system computes an optimal route, which can be further edited and manipulated. In our interface, this route is captured by a parallel coordinate data display in which the dimension ordering is configured by the specified route. Our framework serves both as a data exploration environment and as an interactive presentation platform to demonstrate, explain, and justify any identified relationships.

3.1 MOTIVATION

In the era of information explosion, an important goal for data analysts and researchers is to understand the data collected and make useful decisions from them. More often than not, the most valuable insight comes from intricate inter-relationships among data attributes, and extracting these relationships requires skills in high-dimensional data understanding. Unfortunately, high-dimensional space exceeds human comprehension, and so, effective tools are needed to boost these capabilities. It is well known that graphical displays and visualization can show data patterns more clearly than can plain text and numbers. However, because there is no simple mapping of the multiple dimensions to a two-dimensional screen, more sophisticated visualization techniques than the arsenal of standard plots are needed.

As discussed in Section 2.1.3, one display that is often used for multivariate data visualization is the method of parallel coordinates [60]. It shows the raw data attributes on parallel axes in a sequential fashion. Parallel coordinates plot is good for presenting overviews of the whole, raw data set, as well as for showing relationships among the dimensions. However, the ordering of the dimensions in the parallel coordinate display has a great impact on the visual relationship analysis because the serialization of the data attributes makes it difficult to discern inter-dependencies among non-neighboring axes. As a result, a good dimension ordering can express inter-relationships clearly to the users, while poor dimension orderings can hide relationships potentially of interest. This has already been shown in early work by Bertin [14]. In the worst case, users are faced with the task of trying every possible axis ordering and remembering their findings along the way. Given n dimensions, there are a total of $n!$ possible dimension orderings. But the situation is probably not as dire as that – if we assume that a user can get a good sense of local relationships across 4 parallel axes, then we get $n!/((n-4)!4!)$ which for $n=10$ amounts to 210 orderings – still a daunting number.

Hence, what is sorely needed is an effective interface that can guide users in the selection of promising dimension orderings. Here it is important to note that a number of researchers have proposed possibly tunable metrics computed on the data that can suggest good dimension orderings *a priori* [7][38][64][141][143]. While this alleviates some of these problems, it does not completely eliminate them because there is rarely just one perfect dimension ordering. Especially in interactive data exploration tasks, which are the core of visual analytics, users

should be able to freely navigate the space of promising dimension orderings, and for this they need a suitable "map" of the data landscape.

To address these needs, we describe a network-based interface coupled with an interactive parallel coordinates view for exploring inter-dimensional relationships. It implements the dimension ordering interface as an interactive route planner, operating in a data-informed network where vertices represent dimensions and edges connecting two vertices represent promising dimension pairings. Users can change the route via mouse clicks, and the interface then computes a new optimized route that includes the changed route portion. Other capabilities include multi-resolution navigation and zooming of the network display.

Furthermore, realistic datasets often contain a mix of numerical and categorical variables. Although there are well-defined statistical techniques to handle categorical and numerical variables in isolation, mixtures of these have received less attention. This has a great influence on the sensible computation of correlation factors. How to deal with numerical and categorical variables within a unified framework has been not only a research topic of great interest, but also one of the great challenges in the visual analytics domain.

Finally, causation analysis looks beyond correlation relationship. Causation analysis is an analytical process that tries to answers the questions of causes, reasons, effects, results and consequences. Often it helps analysts make better judgments or predictions by knowing the causes and effects. Here we are focusing on the problem of causation analysis from observational data. Pearl [99] proposed to use graphical model for causal reasoning. The basic idea is to use a *Directed Acyclic Graph* (DAG) to represent the causal relationships, or at least the probabilistic dependencies, between them. In the causal graph, each variable is represented as node. If variable *A* is a direct cause of *B*, then there is an edge going from *A* to *B*. In this case, we also say that *A* is one of *B*'s parents, and *B* one of *A*'s children. If there is a (causal) path from *A* to *B*, then *A* is an ancestor of *B*, and *B* is a descendant of *A*. However, how to build such a causal graph from multivariate data has posed a great challenge for the researchers due to the curse of dimensionality. Moreover, like many other analytical and statistical algorithms, traditional causal model identification algorithms lack of user interactions. The analysts have little influence on the automated analytical process. However, the analysts' expertise should play an important role in the whole process.

3.2 SYSTEM OVERVIEW

Our association mining visual analytics framework uses two coordinated displays: a network display (Fig. 3.1(a)) and a parallel coordinates plot display (Fig. 3.1(b)). Operations in either display are reflected in the other. The network display provides an overview of all the dimensions in terms of the pairwise correlations, whereas the parallel coordinates display shows the raw data with sequentially ordered dimensions.

In the network display (correlation network), vertices correspond to dimensions and are rendered as filled circles. A vertex's size is initially determined by the corresponding

dimension's coefficient of variance, but users may interactively resize vertices as desired. The vertices are laid out via a mass-spring model in which spring rest length is a function of the pairwise Pearson's correlation coefficient. As such, the layout can help users to gain a quick understanding of the correlations existing in the data by comparing the distances of the nodes representing the variables. On the other hand, the placement of the vertices and the edges drawn between them then provide the guidance for navigation and exploration.

An optimal route through the network is computed via an approximate TSP solver – we use a genetic algorithm – and this route determines the order of the dimensions given in the parallel coordinates display. It seeks to place strongly correlated dimensions next to each other. This route is set as the default, but users can re-route using simple interactive mechanisms for specifying which dimensions should be put adjacent. The parallel coordinate display changes the dimension ordering in response to user interaction in the network. Finally, multi-resolution viewing will only show networks involving large vertices.

The default route computed by the TSP solver is particular useful for novice users not familiar with the overall data scenario. They often do not have a clear understanding of all the dimensions and the variables they represent, and thus, given a data set, do not know where and how to start the exploration. The automatic ordering seeks to maximize the correlation that the sequential ordering in the parallel coordinate display could provide. Then based on this ordering, users can interactively assign some constraints in the network display or directly drag and drop axes to change the ordering in the parallel coordinates display manually.

3.3 APPROACH

In the following we describe the various components of our framework and the user interactions they allow. We illustrate each facet with examples derived with the following three datasets:

- Cars dataset [156]—392 cars and 7 attributes: MPG, #cylinders, horsepower, weight, acceleration time, year and origin.
- Automobile data—198 data points and 12 attributes: make, door, body style, wheels, length, width, height, weight, cylinders—, horsepower, MPG, and Price.
- Global Seawater Oxygen-18 dataset [118]—25,476 data points and 8 attributes: longitude, latitude, month, year, depth, temp, salinity, and d18O.
- University dataset—this dataset consists of 50 colleges and 14 attributes: *academics, athletics, campus housing, night life, safety, transportation, weather, dining, PhD/faculty ratio, population, household income, USNews score, tuition,* and *location.* The dataset is an amalgamation of data obtained from two different sources: the College Prowler website [158] and US News & World Report [159]. The former ranks each school across the 20 most relevant campus life attributes. We took the top 50 colleges from US News and three attributes *USNews score, tuition,* and *location*. All the other attributes are from from College Prowler.

- Synthetic dataset—1,000 data points and 25 attributes with several attributes that behave very similarly.

We make use of Pearson's correlation coefficient in a number of ways in our work. Correlation is a statistical technique that can show whether and how strongly pairs of variables (dimensions) are related. Pearson's correlation coefficient between two variables x and y is defined as follows:

$$r(x,y) = \frac{\sum_{i=1}^{N}(x_i - \mu(x))(y_i - \mu(y))}{\sqrt{\sum_{i=1}^{N}(x_i - \mu(x))^2}\sqrt{\sum_{i=1}^{N}(y_i - \mu(y))^2}} \quad (3.1)$$

where x and y are two vectors of the same size, $\mu(x)$ and $\mu(y)$ are their respective means, and N is the number of data points. The correlation, r, ranges from -1 to $+1$. The closer r is to -1 or $+1$, the more closely the two variables are linearly related, whereas r close to 0 means that there is no linear relationship between the two variables

The network is then built as follows (see Fig. 3.2):

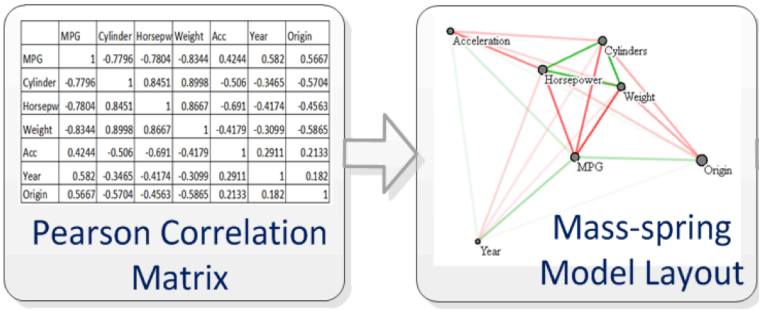


Fig. 3.2: The network generation pipeline.

1. Compute Pearson's correlation for each pair of dimensions.
2. Layout the points using a mass-spring model. The forces between dimensions are computed from the correlations – highly correlated dimensions will be placed close to one another.

3.3.1 Network Construction

The vertices v in the network are positioned using a force-directed layout scheme. Vertices correspond to mass points, and graph edges are springs. The rest length of a spring is determined by the correlation coefficient of the two dimensions represented by the spring (i.e., a strong correlation corresponds to a short rest length).

In the interest of simplicity and efficiency, we employ an explicit integration method to drive the simulation. The new state of the system is determined based on the immediately preceding state of the mass-spring network. The accelerations $\ddot{v}$ of the vertices are calculated by $\ddot{v} = f/m$

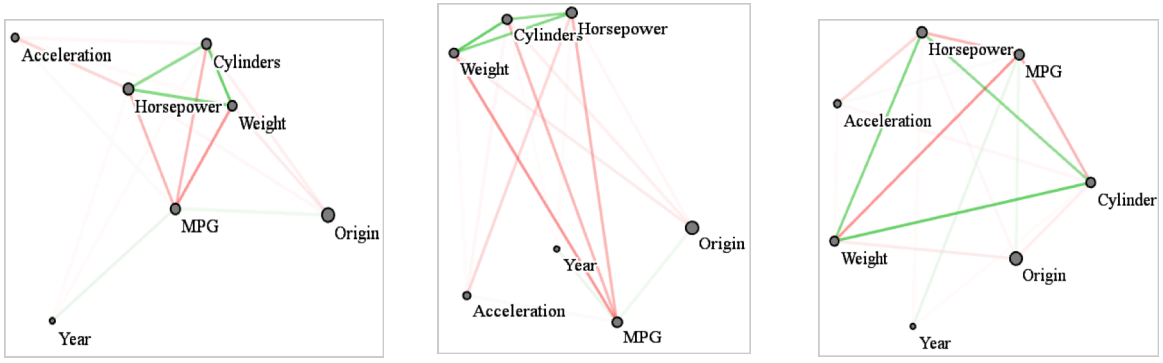
where the force f is computed using Hooke's Law (f=-kx, with k being the spring constant and x being the displacement of the spring's end from its equilibrium position). The vertices are then moved to their new positions by $\dot{v}_{i+1} = \dot{v}_i + \ddot{v}_i \Delta t$ and $v_{i+1} = v_i + \dot{v}_i \Delta t$, where $\dot{v}$ denotes a vertex's velocity, subscripts denote time and Δt is the time-step. The mass (m) of the points, spring stiffness values, time-step and other simulation parameters can be easily determined experimentally so that the simulation remains stable and converges quickly. The simulation is allowed to run until the maximal displacement of any vertex from one time-step to the next is less than a small constant – we chose 0.001% of the rest length of a spring representing a correlation coefficient of 0 (i.e., the maximum possible rest length of any spring).

3.3.2 NETWORK SEMANTICS

The size of a network vertex encodes its significance. In statistics, if a dimension variable is more diverse, it is more interesting to drill down into. As such it potentially plays a more important role in the dataset. We use a diversity measure to encode the significance of a dimension. There are several ways to encode diversity, such as range, standard deviation, or coefficient of variation. We chose the coefficient of variation as the default because it is a normalized and comparative measure of dispersion of the distribution. However, users may choose any of these metrics, build new metrics, or alter the significance manually according to expertise.

The coefficient of variation is computed as: $c_v = \sigma / |\mu|$, where σ is the standard deviation and μ is the mean of the corresponding dimension. Significance is visually encoded as the vertex radius so that more significant dimensions are represented as larger vertices, whereas less significant ones are encoded with smaller radii.

Conversely, the edge significance is weighted by the correlation between the two dimensions linked by the edge. The correlation is encoded by color. Green encodes a correlation value of +1 while red encodes correlation value of -1, and 50% gray is used to encode 0. Linear interpolation computes the colors in between. We provide three correlation functions – correlation strength,



(a) (b) (c)

Fig. 3.3: Dimension layout in the Cars dataset via a mass-spring model. Higher color saturation on edges represents high correlation values. Green represents positive correlation, while red represents negative. (a) Layout with absolute correlation strength; (b) layout with positive correlation; (c) layout with negative correlation preference.

and positive and negative correlation – to help users to comprehensively understand various dimension relationships. Fig. 3.3 provides illustrations using the cars dataset as an example:

(i) Correlation strength (see Fig. 3.3(a)): Here the absolute value of the correlation is used to compute the rest length. Thus, high correlations correspond to shorter rest lengths. In this way, highly correlated dimensions will be drawn towards each other, whereas weakly correlated dimensions will repel each other. In the parallel coordinate display, these strongly correlated dimensions will likely be arranged next to each other by the TSP router.

(ii) Positive correlation (see Fig. 3.3(b)): Here a simple linear transformation of the correlation value, $(r_{XY}+ 1)/2$, is used, where r_{XY} represents the correlation between dimensions X and Y. Now, positively correlated dimensions will tend to be drawn towards each other, whereas negatively correlated dimensions will repel.

(iii) Negative correlation (see Fig. 3.3(c)): This is the opposite case of the positive correlation preference. Now $(-r_{XY} + 1)/2$ is used to compute the rest length.

Thus, since the vertices are laid out by functions of correlation, depending on which function the user chooses, the user can learn quickly which dimension pairs have what type of correlation. For example, if vertices are laid out by strength of correlation (absolute value of correlation), then close dimensions have strong correlations while far-away dimensions have weak correlations. The edge's color then reveals whether it represents a positive correlation (green) or a negative one (red). Similar insight emerges from the positive/negative correlation-preferred layout. This display is helpful since the parallel coordinate display makes it difficult to discern correlation information if two dimensions are not adjacent to each other – the network view readily solves this problem.

3.3.3 NETWORK-DRIVEN DIMENSION ORDERING

As mentioned, our approach conceives an ordering of the parallel coordinates dimension axes by specification of a path in the correlation-based network display. The shortest path covering all vertices is then one that maximizes the amount of correlation exposed by the parallel coordinate display. Finding such a path is the well-known TSP problem. We have chosen a genetic-algorithm- based TSP solver since it defines a maximum time bound for computing the solutions. Users can modify this time bound to strike a balance between performance and accuracy. In our implementation, we set the time bound to 1s which makes the system sufficiently responsive for interactive exploration. We found that when the number of dimensions $n<30$ the routing is reasonably accurate. Conversely, although the performance is dependent both on the number of vertices and the vertex locations, for $n>30$ with the given time bound of 1s, the solution is sometimes not accurate. Here, according to our experiments, an edge-length-based greedy algorithm typically produces better results at a much reduced time. We thus define a threshold $d=30$. When $n<=30$ we use a genetic algorithm while for $n>30$ we run the greedy algorithm.

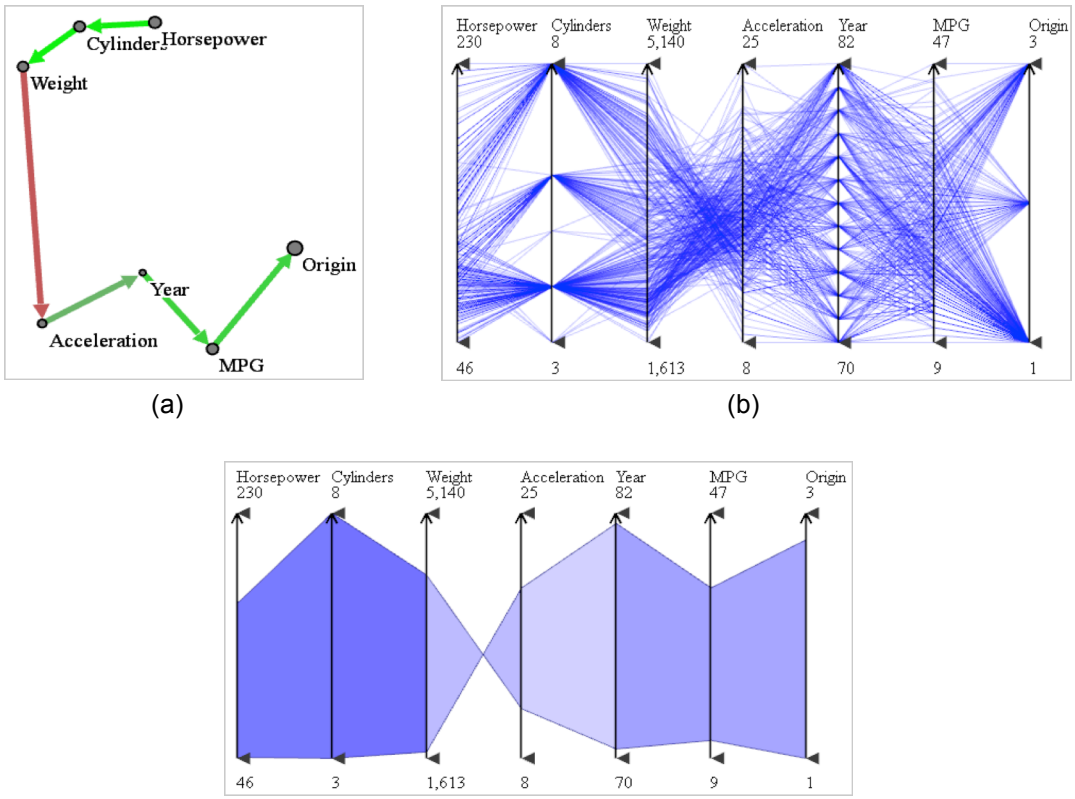


Fig. 3.4: Network-driven dimension ordering. (a) A tour of the Cars dataset. In the network display, the dimension ordering is given as a series of directed edges color-coded by correlation value. (b) Parallel Coordinate display. The dimension triples (*Horsepower*, *Cylinder*, and *Weight*) and (*Year*, *MPG*, and *Origin*) are put next to each other in the parallel coordinate display because they are strongly correlated with each other. These strong positive correlations can be seen on the network display also. (c) Correlation display colors the outlines of line bundles in terms of their correlation strength (less saturation maps to lower correlation). We can also discern negative correlations by the characteristic bow-tie shapes.

Fig. 3.4(a) shows a route within the network display of the Cars dataset and Fig. 3.4(b) shows the associated parallel coordinate display with the dimension ordering implied by the route. The experienced reader will notice that the correlations revealed by the line structure of the parallel coordinate display are quite similar to those visualized by the vertex distances and edge colors in the network display. Of course, the latter is much easier to discern for less experienced users.

To aid these inexperienced users in the visualization of correlations also within the parallel coordinate display, we have devised mechanisms that add additional illustrative hints. These techniques are inspired by McDonnell's illustrative rendering [87]. First, a bounding hull of the line bundles is computed based on the line centers and standard deviations. The difference now is the bounding hull we use for negative correlated dimensions. If two dimensions are negatively correlated, instead of using a band-shape, the characteristic bow-tie shape is employed. Then the bounding hull is colored in terms of their correlation strength where less saturation maps to

lower correlation. Fig. 3.4(c) shows an example for this scheme. We observe that the coloring maps nicely to the vertex distances in the network display.

3.3.4 *Interactions with the Network*

To be effective and appealing for mainstream application, interactions should use paradigms and metaphors that users are already familiar with. For this reason, we employ a route planning paradigm that resembles those found in Web-based, interactive maps.

Our interface allows users to interactively assign constraints to modify the route for data exploration by simple mouse interactions. Such constraints include specifying edges that should to be maintained or avoided on the route, as well as vertices that should be avoided. If an edge is marked to be included in the path, then all paths that do not pass through the edge will be penalized. Conversely, if an edge is marked to be avoided, then all paths that pass through this edge will be penalized. If a vertex is marked to be avoided, then we remove the vertex from the TSP computation. Every time a user makes modifications of this nature the TSP algorithm is rerun to produce a new optimal ordering observing these constraints. Finally, users are also able to specify at which dimension the route starts. In this case, the TSP-generated paths will contain only those starting with the specified dimension.

It can sometimes be useful to duplicate a dimension in the parallel coordinate display to

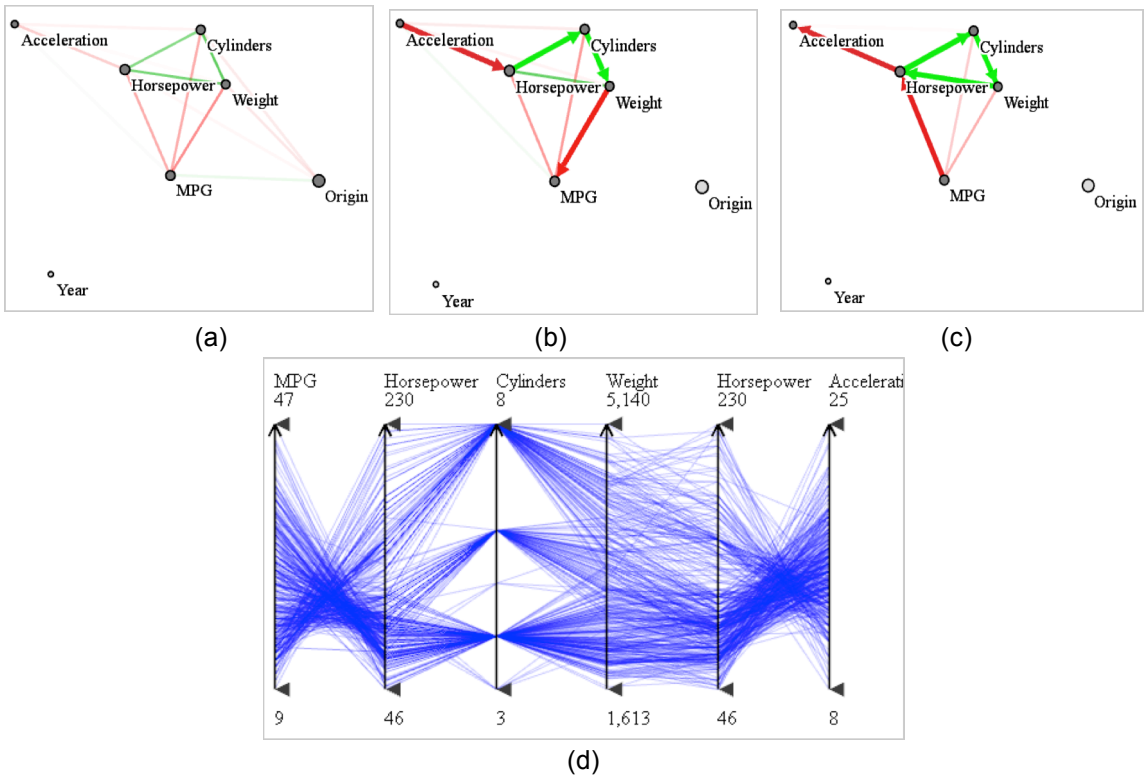


Fig. 3.5: User interaction and constraint imposition for the Cars dataset. (a) Network Display with Year being filtered out by multi-scale zooming. (b) The route is edited to avoid unrelated dimensions *Year* and *Origin*. (c) Routing with a cyclical constraint. Dimension Horsepower is visited more than once, which makes it adjacent to 4 dimensions (*MPG, Weight, Cylinders*, and *Acceleration*). The corresponding parallel coordinate display is shown in (d), where we can see that *Horsepower* has been duplicated.

visualize its interaction with two different variable pairs at the same time. We support such orderings by allowing users to specify a *loop constraint* on the TSP path. For example, a user might be interested to examine variable 2 with variables 1 and 3 but also with variables 4 and 5. One of these pairs will then form a loop and the other will be part of the regular path. To identify the best configuration we run TSP multiple times. In our example, one possible outcome might be that variables 1, 2, and 3 are part of the path and variables 4 and 5 form the loop centered at variable 3, but the opposite can also be the result. This mechanism extends to multiple loop constraints.

Finally, users can also modify the significance of the vertices via a *significance slider*. This slider controls the size of the vertices that are taken into account for the routing. Extending the scale ignores smaller vertices both in the network and for the auto-routing. The ignored dimensions will then be removed from the parallel coordinate display accordingly. A similar mechanism controls the inclusion of insignificant edges – edges with lower correlations strengths – into the TSP calculations.

Fig. 3.5 shows some snapshots of such an interactive routing session, again using the Cars dataset. First, the user enacts the significance slider to filter out the less significant attributes (the attribute *Year* in this case), as shown in Fig. 3.5(a). The user then observes that *Origin* appears to be uncorrelated with (i.e., is distant from) the remaining attributes. So he removes it from the route (Fig. 3.5(b)) to yield a more compact parallel coordinate display (not shown). Further, he also observes that *Horsepower* is highly correlated with both *#Cylinders*, and *Weight* (but also negatively correlated with *MPG* and *Acceleration*). In order to see all of these relationships conveniently in the parallel coordinate display he adds a loop to the route, interspersing *Horsepower* between both relationships and also visualizing the strong 3-way relationship in the loop (Fig. 3.5(c) and (d)).

3.3.5 FOCUS + CONTEXT BROWSING

One problem of the network display (correlation network) is that the visual clutter arises when the number of dimensions grows. Since the correlation network is a complete graph, the high dimensionality makes it difficult to follow the edges [47]. We provide several options to help reduce clutter, following the visualization mantra: overview first, zoom and filter, then details-on-demand [121].

We begin by providing two correlation filtering operators for vertices and one for edges. The filtering operators for vertices are used to control two objective parameters—accumulated correlation R and standard deviation σ—which are controlled by two sliders defining thresholds δ_R and δ_σ. Only vertices with $R_i > \delta_R$ and $\sigma_i > \delta_\sigma$ are shown in the map. The filtering operator for edges allows users to define a threshold δ_e. For correlation strengths (absolute value) smaller than δ_e, the corresponding edges will be filtered out from the correlation network, as shown Fig. 3.6(b). These filtering operators help users guide their focus on highly correlated variables only. In addition, after the mass-spring model layout, the relative locations of vertices can already give indications of the correlation strength information among variables. Thus, users can set $\delta_e = 1$ to

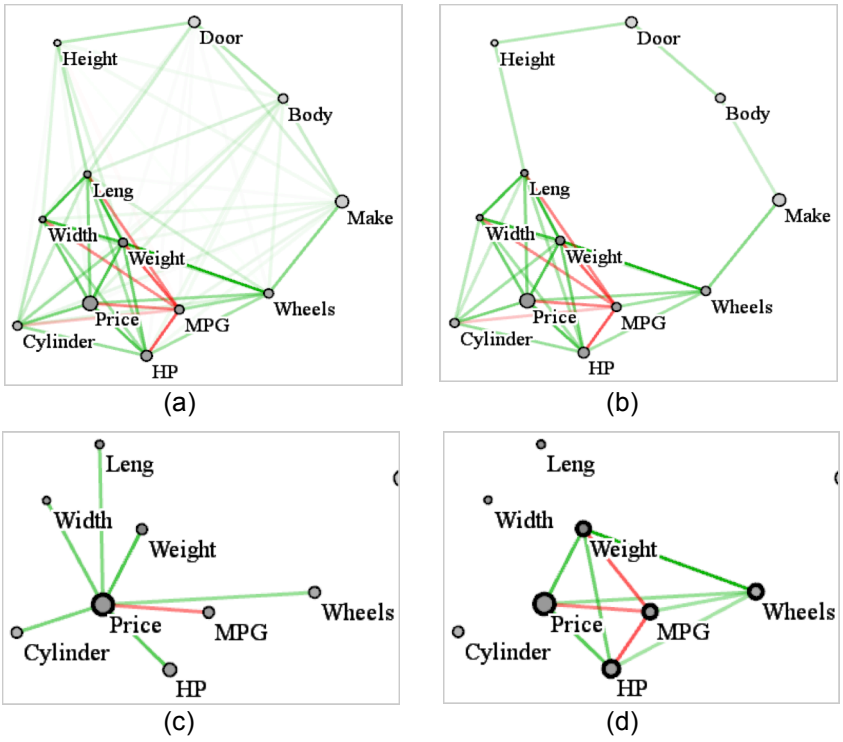


Fig. 3.6: Focus+Context browsing. (a) Correlation network for the automobile dataset. (b) The edge correlation filter δe is set to 0.5 to remove low correlation edges. To get (c) and (d), first the user filters out *Height, Body, Door* and *Make* due to small accumulated correlations. (c) The vertex-browsing correlation network with focus only on *Price*. All edges that are incident on *Price* are highlighted. (d) The edge-browsing correlation network after selecting a group of variables (shown with thicker outlines). Only correlations (edges) within the group are shown to help the user to focus on the variables he/she is interested in.

turn off all edges without losing much correlation strength information. This edge-less correlation network can provide a good overview of the correlations. By default, all thresholds are set to 0, and thus no filtering is applied.

Details-on-demand is supported by two interactive different browsing modes: *vertex-browsing* mode and *edge-browsing* mode. In *vertex-browsing* mode (Fig. 3.6(c)), hovering over a vertex highlights all of its adjacent edges, and clicking on a vertex keeps the adjacent edges highlighted. In *edge-browsing* mode (Fig. 3.6(d)), the user can select a group of interesting vertices; only edges with both adjacent vertices inside the group will be highlighted. In both modes, the selected vertices are rendered with thicker outlines to distinguish them from others. These interactions can help users to interactively explore the correlation space for interesting discoveries.

Let us look at Fig. 3.6(c) as an example. The user has interactively selected *Price* (see lower-left corner) as the focus variable and has set the accumulated correlation filter δ_R to 0.2. As a result, the variables *Height*, *Body*, *Door*, and *Make* are filtered out; only the edges adjacent to *Price* are highlighted to better reveal how *Price* is related to the other variables. Conversely, the

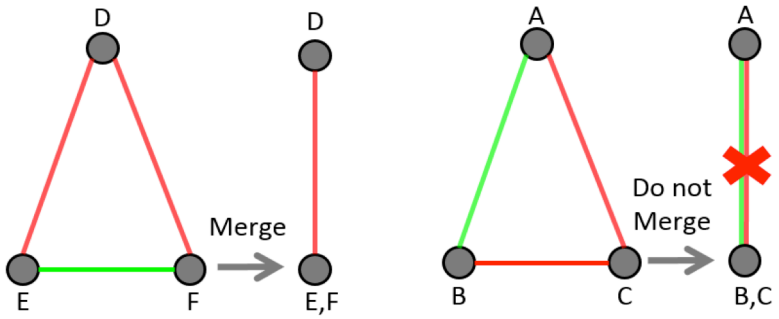


Fig. 3.7: Multi-scale zooming merging criteria. (a) Variables *E* and *F* are positively correlated. As a result, both of them have the same correlation type with variable *D*. Then, when we merge *E* and *F*, the merged vertex retains the same type of correlation as before: *edge(D, EF)* vs. *edge(D, E)* and *edge(D, F)*. (b) Variables *B* and *C* are negatively correlated. Therefore, they have different correlations with variable *A*. If we merge *B* and *C*, the edge from *BC* to *A* will be inconsistent with those before the merging: *edge(A, BC)* vs. *edge(A, B)* and *edge(A, C)*. So we do not merge negatively correlated variables.

other variables with $R_i > 0.2$ are still shown as the context. We can quickly make the following observations:

(1) *Price* is strongly positively correlated with almost all of its surrounding variables, such as *Cylinder, Horsepower, Weight, Length, and Width*;

(2) *MPG* is the only variable that has negative correlations with *Price*. These observations are consistent with our knowledge that cars with high prices (luxury cars) usually have bigger size, different wheel drive (AWD or 4WD), and more horsepower, which results in lower MPG.

3.3.6 MULTI-SCALE ZOOMING

To support the visualization of high-dimensional datasets in limited screen space, a multi-scale framework is required both in the correlation network display and in the PCP display. We provide this functionality via a multi-scale zooming interface similar to what is offered in popular web-based map exploration programs. It requires a similarity metric to construct a hierarchical representation of the variables—the distances between vertices are used to decide whether variables should be merged or not. The merging was motivated by the fact that after mass-spring layout, the distances between pairs of variables provide a global indication of their correlation.

Moreover, after the merging, we do not want to lose correlation information that the display can reveal. Note that two highly positively correlated variables behave similarly in terms of correlations with other variables (Fig. 3.7(a)); while two negatively correlated variables behave differently (Fig. 3.7(b)). When considering merging close variables, we merge only those that have positive correlations. Thus, after the merging, the edges adjacent to the merged variable remain consistent with the original edges. With this merging criterion, as users zoom out of the

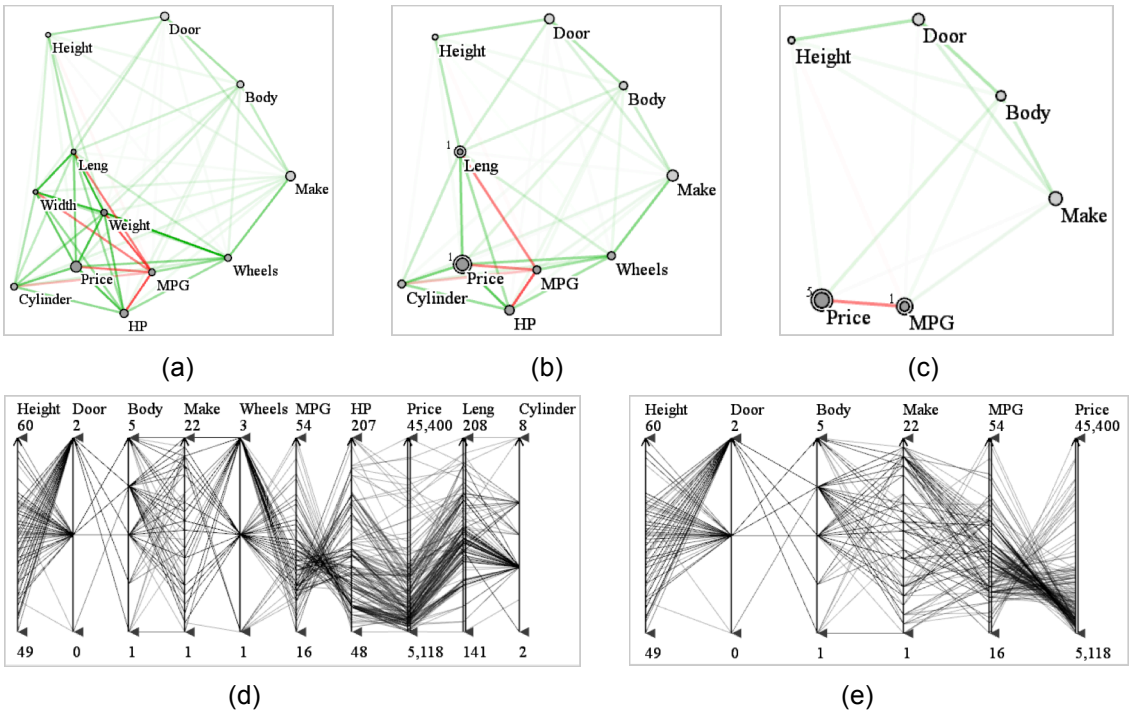


Fig. 3.8: Multi-scale zooming. Original correlation network is shown in (a). As one zooms out, (a), (b) and (c) show the result sequence of views of correlation network. From (c) we can see that variables *Weight, Length, Width, Price, Cylinder, and HP* have been merged into one (*Price*) because all are positively correlated. Although *MPG* is close to them, it has a negative correlation with the others, so it is not merged. But *MPG* and Drive Wheel have positive correlation, so they merge into one (*MPG*). The number of variables packed into the representative variable is given by the small number in the upper left corner of representative vertex. (d) and (e) show the corresponding PCP displays of (b) and (c), respectively. The representative dimensions are shown by double-line axes.

display, nearby variables with positive correlations merge into one; and as users zoom back in, the merged variables split into the original variables.

When considering a representative variable for a set of merged variables, we choose the one with the largest accumulated correlation (R_i). This is justified since R_i not only takes correlation into account, but it is also a data-centric measurement, encoding the variable's significance with regards to other variables. Moreover, the variable with a larger accumulated correlation tends to better indicate or predict the other variables. Another option would be to use factor analysis or PCA to extract the main factor or component as the representative variable. We did not implement this since it is not straightforward to understand what a factor or component actually means, making it hard to interpret relationships from them.

Users can always manually control whether to merge or collapse a representative vertex by mouse-clicks. Fig. 3.8 shows a multi-scale zooming example. Finally, the zooming extends to the PCP display as well—it only shows the representative variables reducing its complexity as well.

3.3.7 *Effect of Parallel Coordinate Display Interactions*

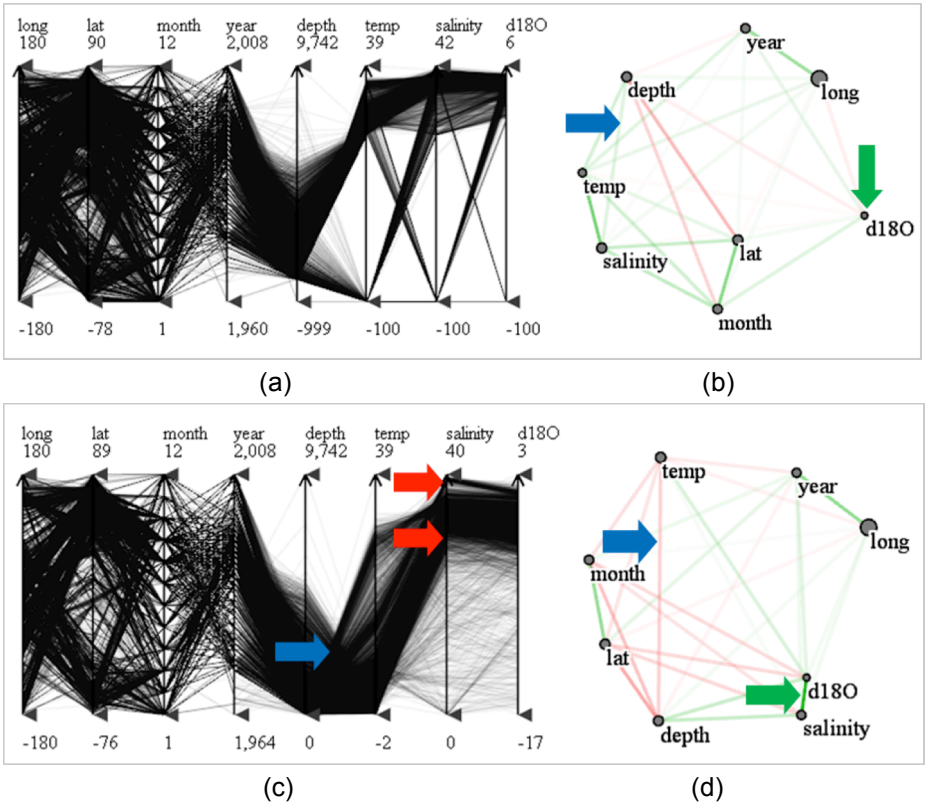
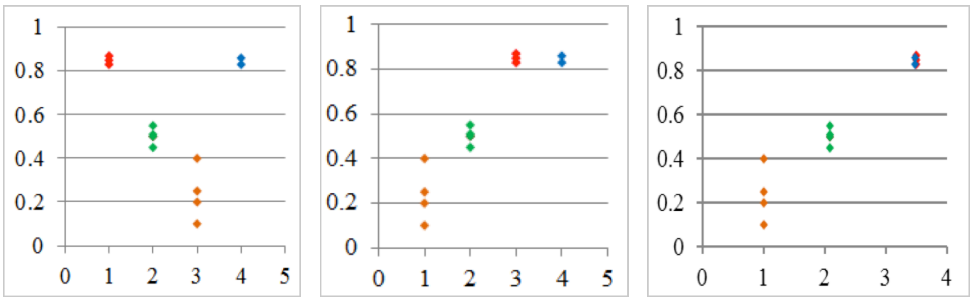


Fig. 3.9: The effect of dimension bracketing, using the global seawater oxygen-18 dataset. (a), (b) Parallel coordinate and network display of the original dataset with undefined values in dimension *depth*, *temp*, *salinity* and *d180*. (c), (d) After filtering out the undefined values and bracketing the dimensions. We can now see a strong positive correlation between salinity and *d18O* (green arrow in (d)), while in the original, unfiltered dataset with undefined values, *d18O* is incorrectly shown not to be strongly correlated with any other dimensions (green arrow in (b)).

Our parallel coordinate display follows the now fairly standard practice of allowing users to filter out undefined values in some dimension (e.g., the undefined values are set to -999) or only visualize a subset of the data falling within a certain range interval of a dimension. For this purpose, our interface provides brush handles for each dimension (see the little triangle on top and bottom of each dimension axis in Fig. 3.9(a)) which can be used to bracket some portions of the corresponding dimension. This "bracketing" prompts the system to filter out the data points that lie outside the handles and only display the remaining data points with proper normalization. All data attributes (e.g., correlation, variance) are then re-computed based on the remaining data.

Let us now demonstrate the effect of parallel-coordinate bracketing and filtering on the co-linked network display. As shown in Fig. 3.9(a) for the Global Seawater Oxygen-18 dataset, the original dataset contains undefined values (-100) for four dimensions: *depth*, *temperature*, *salinity*, and *d18O*. This significantly influences the correlation computation – we observe very weak correlations among all dimensions in Fig. 3.9(b). Following, Fig. 3.9(c) shows the parallel coordinate display after bracketing the four dimensions, filtering out the undefined values and renormalizing their bottom axis brackets – they now show the true minimum values. We may

Location	Crime
A	0.83
A	0.85
A	0.87
B	0.5
B	0.45
B	0.5
B	0.55
B	0.51
C	0.1
C	0.25
C	0.2
C	0.4
D	0.86
D	0.83



(a) (b) (c) (d)

Fig. 3.10: A synthetic data example that shows the necessity of categorical variable transformation in correlation analysis. (a) The original data—the set of crime rates for different spatial locations. (b) Randomly assign a number for each category (A→1, B→2, C→3, D→4). The corresponding correlation is -0.281. (c) Re-order the categories (A→3, B→2, C→1, D→4) while the spaces between categories are fixed (1 in this case). The corresponding correlation is 0.927. (d) Re-order and re-space, (A→3.5, B→2.085, C→1, D→3.48). The correlation is 0.97. The transformation reveals that there is a strong correlation between Location and Crime.

also call this a zooming operation. We can now readily see in the network display (Fig. 3.9(d)) that there is in fact a strong correlation between dimension *salinity* and *d18O*.

3.4 HANDLING CATEGORICAL VARIABLES

Realistic datasets often contain a mix of numerical and categorical variables. Although there are well-defined statistical techniques to handle categorical and numerical variables in isolation, mixtures of these have received less attention. Fig. 3.10 shows an example of this problem for a dataset that lists a set of crime rates for different spatial locations. We can see that the correlation between the two variables increases considerably by simply re-ordering (re-binning) the categories in Fig. 3.10(c) (compared to the ordering in Fig. 3.10(b)), and the correlation becomes even greater when re-ordering is combined with an adjustment of the spaces between categories. This simple example illustrates the need for a more careful optimization of the numerical values chosen for the categorical variables. The next Section is dedicated to this transformation.

A few methods [63][112] employed and extended correspondence analysis to transform either categorical or numerical variables into appropriately spaced and ordered categorical variables. We discuss these methods and their limitations more closely in Section 3. Ma and Hellerstein [85] re-order the categories by inter- and intra-cluster ordering. However, using an equal distance between adjacent categories does not convey the degree of their similarity. In the statistics literature, a popular approach has been to discretize numerical variables into bins and apply statistical methods for categorical variables on them. The inherent problem with this approach—the loss of detail after binning—has been well-reported [27][113][132]. Thus, some

statistics methods encode categories as numerical values [28][135], which enables one to apply statistical methods designed for numerical data. Typically, however, these methods do not consider the ordering and the distances between categories, which are important features for correlation analysis. We provide a method that also maps categorical variables to numerical ones but optimizes the distances.

3.4.1 THEORETICAL BACKGROUND

The methods available for correlation analysis can be divided into three major groups based on their target variables: (1) methods applicable only to numerical variables, such as Pearson's correlation coefficient; (2) methods applicable only to categorical variables, such as Cramér's V; and (3) methods applicable to computing correlations between numerical and categorical variables, such as the t-test, ANOVA, and MANOVA.

Recall Eq. (3.1), Pearson's correlation coefficient is one of the most popular measures for defining linear relationships between two variables:

$$r(x,y) = \frac{\sum_{i=1}^{N}(x_i - \mu(x))(y_i - \mu(y))}{\sqrt{\sum_{i=1}^{N}(x_i - \mu(x))^2}\sqrt{\sum_{i=1}^{N}(y_i - \mu(y))^2}} \tag{3.1}$$

where x and y are two vectors of the same size, $\mu(x)$ and $\mu(y)$ are their respective means, and N is the number of data points. The correlation, r, ranges from -1 to $+1$. The closer r is to -1 or $+1$, the more closely the two variables are linearly related, whereas r close to 0 means that there is no linear relationship between the two variables.

Cramér's V is computed from the χ^2 statistic and can be applied to two categorical variables of any type:

$$v = \sqrt{\frac{\chi^2}{N(k-1)}} \tag{3.2}$$

Here, χ^2 is derived from Pearson's chi-squared test and k is the number of rows or columns, whichever is smaller. The metric v ranges from 0 to 1. The closer v is to 1, the more association the two variables have, while a value of v close 0 means no association between them. v equals to 1 only when the two variables are identical. Similar to Pearson's correlation coefficient, Cramér's V is a symmetrical measure—it does not matter which variable is placed in the columns and which in the rows. Also, the order of categories in the rows/columns does not matter. This makes it an appropriate general-purpose measure.

In comparison, the results from the t-test, ANOVA, and MANOVA, which can handle both numerical and categorical variables, are not normalized. This means that they will have different values given different conditions. Thus, to determine if a relationship is strong or not, one must consult specific significance tables. This makes these measures awkward to integrate into an

interactive application such as ours. Hence, it is best to first transform a pair of mixed variables into a homogenous pair, either both categorical or both numerical, and then apply Cramér's V for categorical variable pairs and Pearson's equation for numerical variable pairs. Next, we describe alternatives to resolve these mixed variable pairs.

3.4.2 Dealing with Mixed Variable Pairs

When dealing with two categorical variables it is often important to define a proper order and spacing of the individual categories (levels) of each variable. The aim is to order and space the levels of one categorical variable with respect to the other, and the essential task here is to gauge the distribution similarity of the levels of the first variable with those of the second variable. Levels with similar such distributions are then spaced closer together and others are spaced further apart. This is a global optimization problem and is commonly solved with Correspondence Analysis (CA). Starting from a contingency table, CA computes the set of independent components—similar to PCA for continuous variables. Then the projective coordinates of the first independent dimension give the transformed numerical values. Rosario et al. [112] devised a method that used CA in the context of parallel coordinates. They then extended the scheme to Multiple Correspondence Analysis (MCA) in which the operations are performed with respect to all categorical variables.

For mixed variable pairs, Johansson et al. [63] proposed to first transform all numerical variables into categorical ones and then use the techniques prescribed for categorical values from thereon. They offered two transformation techniques – an interactive approach tied to a parallel coordinate visual interface and an automated one based on k-means clustering. However, transforming numerical to categorical variables always results in some amount of discretization, unless the objective number of bins equals the set of unique numerical values. But as the number of bins increases, the computation of any subsequent analysis becomes exceedingly expensive. This makes interactive applications difficult.

Given the disadvantages associated with the typical numerical-to-categorical transformation approach, we chose to devise a scheme that goes the other direction—from categorical to numerical. This avoids the problems with binning and enables the use of Pearson's correlation exclusively. It also allows us to find, for a given categorical-numerical variable pair, the spacing and order of the categorical variable's levels that maximize correlation.

Lastly, we need to define the scope of our transformations. We consider categorical variables as variables that have no numerical value, such as gender or occupation. These are the variables generally defined as categorical variables in statistics. On the other hand, since our approach possibly reorders the levels, we cannot handle ordinal variables. These are variables with values whose order is significant, but on which no meaningful arithmetic-like operations can be performed. To cover them we could constrain our method to only change their spacing which would give some insight on nonlinearities. Even more constraints apply to cardinal and interval

values. These are ordinal variables with the additional property that the magnitudes of the differences between two values are meaningful.

Regression Model for Categorical Variables

As mentioned, the coefficient of determination r^2 is the square of the correlation coefficient, r, and as such ranges only between [0, 1]. It indicates how well the data points fit a line (if it is a linear model). The objective function for least squares regression is the *residual sum of squares* (*RSS)* which is the data variance unexplained by the regression model. The goal is to minimize *RSS* which maximizes r^2 (and therefore r) since r^2=1-*RSS*/*TSS*. *TSS* is the *total sum of squares*—the sum of squared deviations of the dependent variable values from their mean. We note that since we maximize r^2 the correlation factors that result will always be maximally positive. This, however, is no contradiction since we can always reverse the transformed data axis to reverse the sign of the correlation factor as well.

Regression deals with categorical variables via the introduction of *dummy variables*. There is one such variable for each categorical level minus 1. Let us assume we have an independent, 3-level categorical variable and a numerical dependent variable. This results in the following regression model:

$$Y_i = \beta_0 + \beta_1 I_{1i} + \beta_2 I_{2i} + e_i \tag{3.3}$$

where $0 \leq i \leq N-1$, Y is the dependent variable, $I_{1,2}$ are indicator variables (value 0 or 1), $\beta_{0,1,2}$ are the coefficients returned by the regression, and e is a standard error. The indicator variables are only set to 1 when the data point indexed by i is at the corresponding categorical level. The baseline $Y_i = \beta_0 + e_i$ at which neither I_1 nor I_2 is set to 1 is when the third categorical level is active. This model is solved via least squares optimization as usual. Since there is only one dependent variable, this model is called a *univariate multiple regression model*. It can be written in matrix form as $y=X \cdot b$ where y is the $N \times 1$ vector of N observations, b is the $M \times 1$ vector of coefficients with M categorical levels, and X is the $N \times M$ independent variable matrix with indicator values I.

The task we are faced with in our specific problem is to determine the spacing and ordering of the levels of a categorical variable with respect to a numerical variable. Conceptually, this can be accomplished by solving a regression model in which the categorical variable is the independent variable—one dummy variable per level less baseline—and the numerical variable is the dependent variable. The coefficients returned by the least squares optimization then determine the desired order and spacing of the corresponding categorical levels.

Multivariate Regression Model

In our application a categorical variable may participate in more than one pairing with a numerical variable. Using the arguments in the previous sub-section this is equivalent to having

more than one dependent variable. It extends the univariate multiple regression model to a *multivariate multiple regression model*. In matrix form such a model is written as $Y=X\cdot B$ where Y is the $N\times P$ dependent variable matrix with P paired numerical variables, B is the $M\times P$ coefficient matrix, and X is the $N\times M$ independent variable matrix as before. In multivariate multiple regression, each column of B is solved independently and hence there are different dummy variable coefficients for each of the P numerical variable pairings this categorical variable has. The implication for our transformation scheme is that the transformed categorical variable will potentially have different level orders and spacings, one for each of its P numerical variable parings, maximizing the correlation.

3.4.3 TRANSFORMING THE CATEGORICAL VARIABLES

Although it serves as a good theoretical background we do not need to solve a regression model to determine the transformation. We can simply minimize *RSS*. Suppose we are given a dataset Ω with two variables: one categorical variable v_c and one numerical variable v_n. Let us assume N data points and M levels in v_c. Let M^i be the total number of data points that fall into categorical level $v_c(i)$ and let $v_n^i(j)$ represent the j^{th} numerical data point that falls into category level $v_c(i)$. The goal is to transform each categorical level $v_c(i)$ in v_c to numerical values $v_c'(i)$ that maximize r. As discussed in Section 3.4.2, maximizing r^2 (and therefore r) is equivalent to minimizing *RSS* to yield the *RSS* of the transformation, *RSS'*:

$$RSS' = \sum_{i=1}^{M} \sum_{j=1}^{n=M^i} (v_n^i(j) - v_c'(i))^2 \tag{3.4}$$

Letting $\mu(v_n^i)$ be the mean of all numerical data points that fall into categorical level $v_c(i)$ allows a sequence of manipulation of Eq. (3.4) which are outlined in Appendix 1. We then arrive at the following expression that needs to be minimized:

$$\sum_{i=1}^{M} \sum_{j=1}^{n=M^i} (\mu(v_n^i) - v_c'(i))^2 \tag{3.5}$$

Minimization occurs when:

$$v_c'(i) = \mu(v_n^i) \tag{3.6}$$

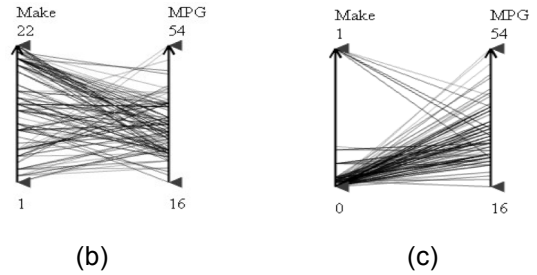
Hence, *RSS'* is minimized for a transformation at which each categorical level $v_c'(i)$ is the mean of the corresponding numerical values falling into it, $\mu(v_n^i)$. This scheme is not only elegant but also computationally very efficient since it only requires the calculations of a set of means.

Lastly, the extension of this method to parings of a categorical variable to multiple numerical variables is straightforward since each pairing can be treated sequentially and in isolation, as was shown in Section 3.4.2.

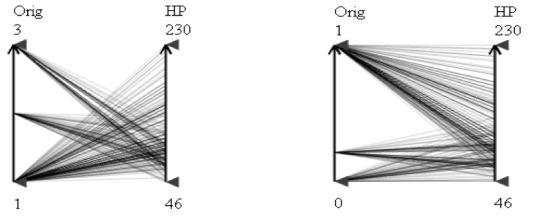
Transformation Results

Dataset	Variable pair categorical/numerical	Corr (Rand)	Corr (Opt)
Auto	make/length	0.115	0.831
	make/price	0.152	0.8871
	make/MPG	0.051	0.712
	make/HP	0.049	0.690
Car	origin/HP	0.034	0.573
	origin/weight	0.112	0.620
	origin/MPG	0.145	0.579
	origin/acceleration	0.054	0.322

(a)



(b) (c)



(d) (e)

Fig. 3.11: Transformation results. (a): Correlation coefficients obtained by randomly assigning an integer value 1 through M to each of the M categories (Rand) and by using the spacing and ordering computed by our optimization (Opt): The correlations achieved by optimization are significantly higher, in many cases by an order of magnitude, for both datasets we tested: Auto and Car. (b), (c) and (d), (e) are two pairs of the parallel coordinate tiles that show the visual improvement after the transformation.

Fig. 3.11(a) shows how this optimization performs, using the Auto MPG (car) dataset [156] (N=398) and the automobile dataset [157] (N=205). In the table, the second column gives the variable pairs and the third column shows the outcome for a random value assignment for each level. The fourth column shows the (greatly improved) correlations obtained with our optimization method. Fig. 3.11(b), (c) and (d), (e) show two pairs of parallel coordinate tiles before and after the transformation, respectively. We can see from the figure that after the transformation, (1) categories (levels) that behave similarly are put close to each other; (2) the correlation is be more visible in the parallel coordinate plots.

3.5 INTEGRATING DATA—THE SUBSPACE SCATTERPLOT

While the adjunct PC display provides access to the raw data, it requires users to transition their eyes to a different screen area. Further, due to PC's sequential axis ordering, multivariate data relationships are difficult to detect across more than three dimensions. Multivariate scatterplots can overcome these problems (see [95] for example). In the following we describe our attempt to integrate multivariate scatterplots into the correlation map.

Our method integrates the data into a tessellation derived from the variable layout. This is another reason why a matrix view is not a feasible option for our system – such a view would not allow a tight data integration. On first glance the visualizations we produce somewhat resemble the hyperbox [5], but we allow scatterplot tiles of more than two variables. Finally, Claessen and van Wijk [25] describe a system that uses tiles of 2D scatterplots and links their axes by the

corresponding parallel coordinates display segments. The tiles in our framework are more general and are directly linked at their shared axes.

3.5.1 TESSELLATING THE CORRELATION MAP

As a first step, we need to create bounded areas into which we can project the data. We accomplish this by tessellating the scattered point set formed by the correlation map's vertices. We use the following strategy:

1. Triangulate the domain using Delaunay triangulation. Delaunay triangulations maximize the minimum angle of all triangle angles in the triangulation. It tends to avoid skinny triangles which are less capable of showing the details of the corresponding subspace.
2. Sort all the edges in ascending order by edge length, which is equivalent to sorting the correlation coefficients in descending order.
3. Optionally, for edges with length less than some threshold, (i.e., a correlation greater than some value), if removing the edge will not cause concave polygons, remove it. Concave polygons are not suitable since our data mapping method requires convex primitives.
4. Create the scatterplot for each of these subspaces using the method outlined in Section 3.5.2.

The third operation will yield polygons with more than three vertices and can be used to visualize data distributions due to higher-order subspaces. Since the variables used to create the polygon are close, the dimensions in these subspaces are sufficiently correlated to give rise to meaningful data configurations in the projections.

3.5.2 GENERATING THE SUBSPACE SCATTERPLOTS

After the tessellation, the map is divided into a mesh of polygons, each due to a data domain subspace. The next step is to project the subspace data into their associated polygons, generating the subspace scatterplots. For this we require a method that can forward project a high-dimensional data point p into the geometry of a concave polygon P defined by S vertices q_i ($0 \leq i \leq S-1$), where S is the dimensionality of p's subspace. The projection of p into P's 2D domain yields a point p' and is a function of the spatial coordinates of the q_i which represent the variables spanning the subspace. In other words, if p's only non-zero coordinate were in variable i, it would map directly to vertex q_i. For all other constellations, the following weighted mapping is utilized, where the weights are the attribute values normalized by the sum of values for all of subspace attributes of p:

$$p' = \sum_{i=0}^{S-1} w_i q_i \qquad w_i = p(i) \Big/ \sum_{j=0}^{S-1} p(j) \tag{3.7}$$

where the $p(j)$, $0 \leq j \leq S-1$, (likewise $p(i)$), are the coordinates of p in its high-dimensional subspace and the q_i are the spatial coordinates of the subspace polygon's vertices in the

correlation map. This mapping is adapted from the method of generalized barycentric coordinates [91]. The coordinates of p are measured with respect to the subspace origin which in our case is the vertex of the subspace (hyper) bounding box that has the minimum value in all subspace dimensions. We note that while the relationships among the projected points are not exactly correlations or cosine similarities, they share some properties of these metrics, such as the insensitivity to scaling in high-dimensional space.

Finally, the color of a point is determined by its cluster membership. The mapped points are organized into pixel-bins in the subspace polygons. We record the maximum/minimum extent of the S-dimensional bounding box of each point cluster, use it to determine density, and indicate denser bins by higher intensity. There might be cases in which some regions have very high density while most other regions have low densities. The low density points will then become difficult to see after intensity normalization. We provide a slider bar that allows users to control the degree of transparency of the scatterplot. If the value is 0, all points will have the maximum intensity, that is, there will be no transparency at all.

3.5.3 *Reading the Subspace Scatterplots*

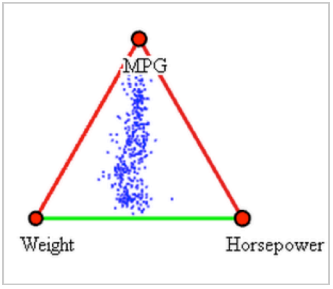


Fig. 3.12: Subspace scatterplot.

The subspace scatterplot generalizes RadViz [57] from a circle to a generalized polygon. Similar to Radviz, the location of a projected point indicates how much it gravitates towards a particular attribute (or set of attributes). This allows the assessment of biases, trends, and trade-offs. For example, in Fig. 3.12, we observe a positive correlation between *Weight* and *Horsepower* and a negative correlation between *MPG* and both *Weight* and *Horsepower*. In the corresponding scatterplot we observe that all cars map to a long cluster centered between *Weight* and *Horsepower* and reaching towards *MPG*. This effectively visualizes the trade-off that exists between weight and horsepower—there are no light cars with high horsepower and vice versa – and it also shows that high MPG requires one to lower both weight and horsepower but that this trade-off function is smooth and continuous.

3.6 Using the Network for Story Telling with Data

The parallel coordinate display provides a sequential view of the data. Reading the plot from left to right is similar to reading a story from beginning to the end. Of course, just like in parallel coordinates, readers of a book or viewers of a movie may skip back and forth to recap what has already been seen or peek ahead what is yet to come. Directors of movies commonly use story boards to organize the shots into a suitable sequence. Here one typically aims to arrange a sequence that builds the story in a coherent manner, with some sort of climax in the end. With

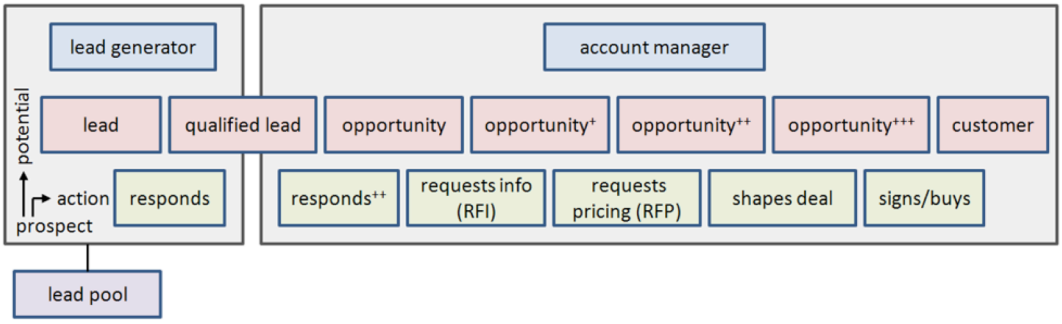


Fig. 3.13: Anatomy of a sales pipeline

'coherence' one might define that subsequent scenes bear some degree of correlation (yet we admit that movies exist that have turned the lack of coherence into an art form).

We propose to utilize our network display as a means for story boarding in information visualization with parallel coordinates. Our network display provides all needed information and functionality to help visual analysts script insightful and informative 'movies' in parallel coordinates. It reveals correlated shots (i.e. dimensions) and it allows automated and manual interactive arrangement of these shots, using the network interaction facilities.

We shall demonstrate the use of this interface via our sales campaign dataset. It consists of 900 data points (one per sales person) and 10 attributes: *%Completed, #Leads, Leads Won, #Opportunity, Pipeline Revenue, Expected ROI* (Return on Investment), *Actual Cost*, *Cost/WonLead, Planned Revenue,* and *Planned ROI*. There are three pre-clustered sales teams.

Before delving into this case we need to review some basics, as shown in Fig. 3.13. The typical corporate sales pipeline begins with a *lead generator* whose job is to produce a number of prospective customers that have some level of probability that a salesperson will actually close a deal with them. Upon a positive response, these *leads* become *won leads (qualified lead)* and receive an increased sales pitch at a *cost per won lead*. If this pitch wins further positive response, then these won leads become *opportunities*, which might be potential customer in the future. In practice there are many more levels, but this may serve as a sufficient model here. Of course, money or cost is involved in each step of the pipeline, which is another important factor that should be considered.

As a practical scenario let us imagine a meeting of sales executives who would like to review the strategies of their various sales teams. The data contains three sales teams of a large corporation with a couple of hundred sales people in each team. Jim, one of the sales strategy analysts begins and constructs Fig. 3.1(a) (page 12). This display reveals that the *#leads, #won leads, #opportunities*, and *cost/won lead* are somewhat related. The TSP computes an initial route that gives rise to the parallel coordinate display in Fig. 3.1(b). Jim quickly notes that this route does not really represent the actual flow of a lead through the sales pipeline and changes the route to *#leads* → *#won leads* → *#opportunities* (not shown). Soon after, Kate, another sales analyst in the meeting room, realizes from looking at the network display that *cost/won lead* is

nearby and has a strong positive correlation with *#opportunities* but also a negative correlation with *#won leads*. She suspects that some insight could possibly be gained from routing these two latter variables through the first. So she uses the mouse and designs the route shown in Fig. 3.1(c) which gives rise to the parallel coordinate display of Fig. 3.1(d). From the parallel coordinate plot it is now immediately obvious that the blue team employs a very different strategy than the green and the red teams. The blue team generates far fewer leads but spends much more resources on each which apparently gives it an advantage in the final outcome. It can also be observed that the blue team is much more consistent than the other teams, as indicated by the much narrower band.

3.7 CORRELATION ANALYSIS

Correlation analysis looks for relationships between variables and can show whether pairs of variables (attributes, dimensions) are related and how strongly. It has become increasingly popular in psychology, education, finance, marketing, and climatology, just to name a few. Correlations, however, are difficult to interpret, manage, and survey once the number of attributes becomes even moderately large. Given D variables, there are $O(D^2)$ correlation pairs, which makes complex relationships difficult to recognize from columns of numbers alone. Hence, there is a clear need for an effective visual interface that allows analysts to (1) quickly get an overview of the overall correlation relationships in the data, and (2) easily manipulate the data to reveal hidden relationships via different modes of interactions, such as filtering, selection, and clustering.

Correlation invokes a sense of neighborhood—attributes that are more tightly correlated are perceived as being closer. Hence, the network display we advocate in this chapter is well suited for its visualization. As mentioned, the correlation network comes in handy to assist the analysts in finding the most relevant ordering. Conversely, parallel coordinate plots allow bushing and filtering to focus on certain sub-patterns of interest which can be used to reconfigure the correlation network. It is these mutual benefits that suggest a tightly integrated framework of the two constituents: correlation network and parallel coordinates. We demonstrate how we use our framework to help with correlation analysis problems.

3.7.1 CORRELATION ANALYSIS: THE UNIVERSITY DATASET

We first demonstrate the basic concepts of our framework with the university dataset. Fig. 3.14(a) shows the correlation network. We observe that *location* is quite strongly correlated with a number of variables nearby, such as *night life*, *safety*, *transportation* and *population.* The last a ttribute is particularly interesting in that its large vertex size indicates that the dataset contains a large variety of university settings—urban, suburban, and rural.

From Fig. 3.14(a) we also observe that the majority of correlations are not overly strong, as is apparent from the mildly saturated edges and vertices. So in order to isolate the more significant correlations we raise the edge correlation threshold to δ_e=0.3. The resulting map is shown in Fig. 3.14(b) where we observe two fairly independent clusters—one dealing with academic aspects,

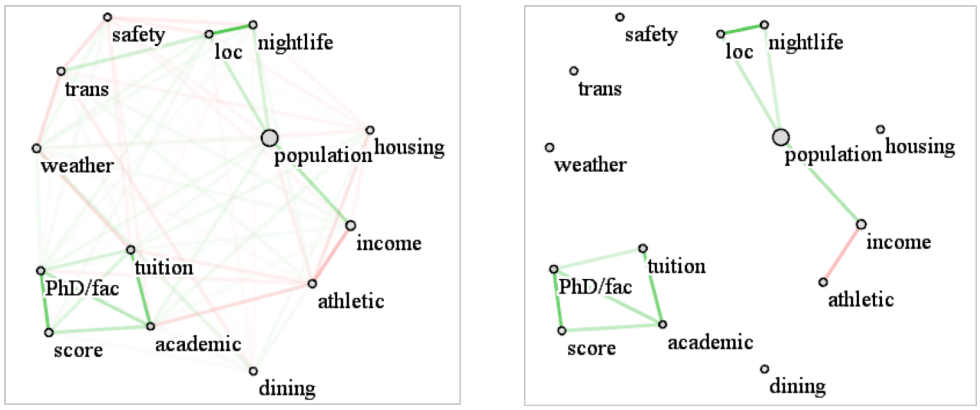


Fig. 3.14: Correlation network for the University dataset. (a) Original correlation network. (c) After applying the edge correlation filter (setting δe=0.3) – the data divide into two fairly independent clusters – one (lower left) dealing with academic aspects, the other (upper right) with student life.

the other with student life. This reveals that these two aspects of the college experience tend to be largely independent in general.

We also observe that the correlations within either of these clusters are mostly positive (indicated by the green edges). In the 'academic' cluster at the bottom left of Fig. 3.14(b) all variables (*US News Score, PhD/Faculty ratio, Tuition,* and *Academics*) are positively correlated with one another. Hence, when one variable increases, all others will increase too, and vice versa. This observation is consistent with our knowledge that highly ranked universities (high *US News Scores*) usually have better *academics* and higher *tuition*. Yet, because students are more willing to go there, the *PhD/faculty ratio* is higher.

On other hand, in the 'student life' cluster on the right of Fig. 3.14(c) we observe that *athletics* is negatively correlated with *income*, whereas *income* is positively correlated with *population*. A possible explanation for this is that the universities with good athletics are usually located in rural areas, which are less densely populated, and the income in these areas is relatively low compared to other more populated areas (e.g., New York City). We also find that *night life* has a high positive correlation with *location* and *population*, which is also justifiable.

Many more conclusions can be drawn from this single visualization, and so we believe that these maps can be helpful for students to select universities, as well as for university executives to make policies.

3.7.2 CORRELATION ANALYSIS: THE SALES CAMPAIGN DATASET

We use the sales campaign dataset to show how our framework can help business executives in making marketing decisions. The basics of the dataset are discussed in Section 3.3.

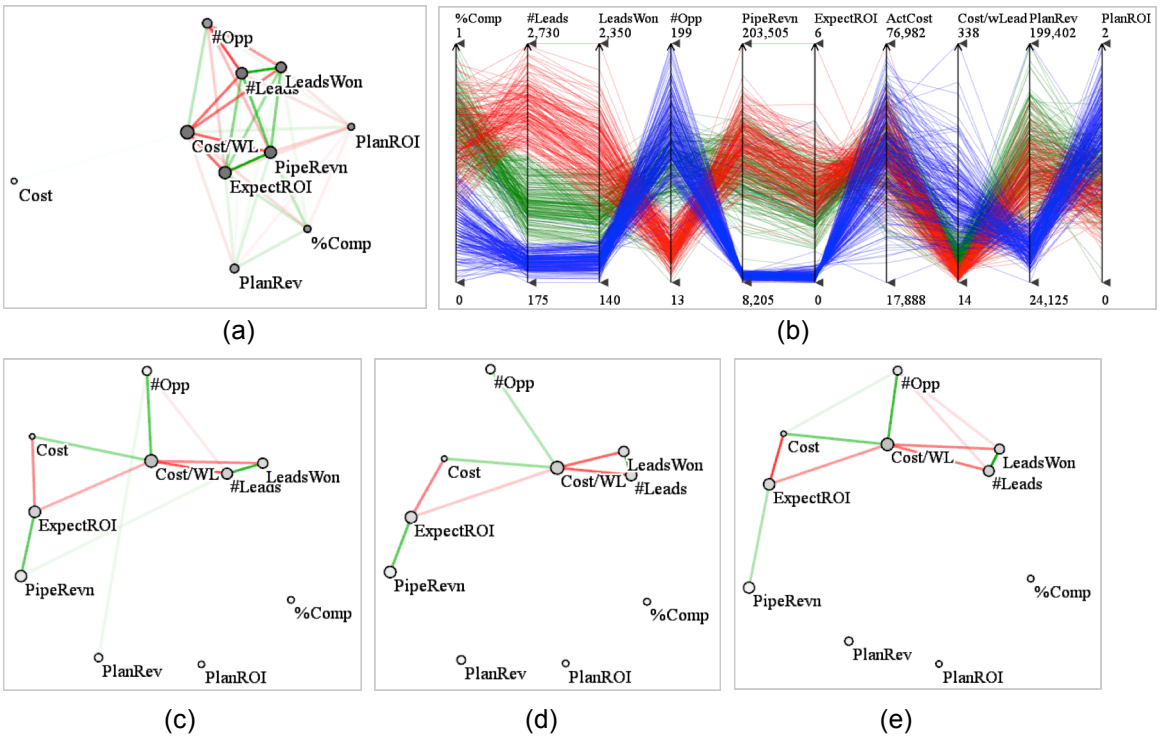


Fig. 3.15: Business strategizing with the sales campaign dataset (a) Aggregate correlation network for all three sales teams. (b) Parallel coordinate plot for the three sales teams (clusters), colored red, green, and blue. (c)-(e) Correlation networks for the red, green, and blue teams, respectively. All edges with correlation strength less than 0.2 have been filtered out to extract the main structure of the map.

As a practical scenario, let us imagine a meeting of company executives who would like to make sales policies for the next year based on their three sales teams' behaviors of this year. John from the marketing department always wants more opportunities. By looking at the correlation network of the three teams (Fig. 3.15(a)) he states that since *cost* does not have strong correlations with other variables, the company can make any strategies for other variables, and it will not influence the actual cost. So, he proposes that the company should improve efforts to create more *opportunities* for the next year without considering the money issue. Based on the correlation network, such efforts could be reducing the number of *leads* and *won leads*, thus increasing the *cost per won lead*.

However, Emily, from the financial department, believes that there must be something wrong with this statement since *cost* should play an important role in the sales pipeline. By looking at the data space, the PCP plot, which is shown in Fig. 3.15(b), she notices that these three sales teams behave quite differently. It is likely a mistake if they consider the three teams together. Hence, she suggests that they plot the correlation networks for the three teams separately. The results are shown in Fig. 3.15(c), (d), (e), for the red, green, blue teams, respectively. It is interesting to note that the three teams have quite similar correlation patterns, which is consistent with her expertise that there must be some marketing model that guides the sales behaviors and the model should involve *cost* in it. From the plots, one can see that there are 7 variables

involved in the pattern: *opportunities, cost, cost per won lead, lead, lead won, expected ROI,* and *pipeline revenue*; other variables are not as closely related. As a result, these 7 variables should be focused on as references to make decisions.

Based on these observations, Emily claims that the actual influences of increasing the *opportunities* should be: (1) *cost per won lead* will be increased because it is the only one that is related to *opportunities* in the plot, with positive correlation. However, *cost per won lead* is highly correlated with four other variables. As a result,

Based on those observations and Emily's own expertise on causal factors and their relationships, Emily interprets correlation as true causation and claims that the actual influences of increasing the opportunities should be (1) the *cost per won lead* will be increased because it is the only variable that is directly related to opportunities, with positive causation; (2) the number of *leads* and *won leads* will be decreased due to the negative causation; (3) the *cost* will be increased and the *expected ROI* will be decreased. So she proposes to reduce the *cost* for the next year. The corresponding impacts, according to Emily, (1) the *expected ROI* will be increased due to its negative causation with *cost*, which is another good factor; (2) the *cost per won lead* will be reduced due to its positive causation with *cost*; (3) the *opportunities* will also be reduced which is a negative effect. After listening to these two proposals, CEO Tom is about to make final the decision. First, increasing *cost* is not preferred because this year's expense already exceeds the budget. Second, based on Emily's expertise, her causal interpretation of the situation, and her assessment based on the resulting causal inference, although the number of opportunities is reduced, the *expected ROI* will go up. By considering these conditions, Tom decides that the policy should follow Emily's proposal.

3.7.3 DISCUSSION

Correlation analysis is useful for exploring the relationships between pairwise variables and for predicting one variable's behavior based on another, as shown in the previous sections. However, correlation analysis also has its disadvantages. The main drawback is that a relationship between two variables does not imply a causal effect: any two variables could be correlated, but this does not necessarily mean that one is the cause of the other. One example could be the automobile dataset (Fig. 3.6(d)). *Price* has a strong positive correlation with *Weight*. We can only say that given a high price, usually we can predict that the car has high weight, or vice versa. But we cannot say that high price causes high weight, or vice versa. The other drawback of the correlation is that it applies only to variable pairs. Sometimes we need multiple factors to explain one behavior. Nevertheless, we note that the purpose of our framework is not to provide innovations in addressing limitations of correlation analysis, but to provide an efficient tool to help analysts to do interactive correlation analysis and predictions.

3.8 SUBSPACE SCATTERPLOT BASED ANALYSIS

Manipulating the subspace scatterplots can also reveal many interesting relationships, in situ with other information in the correlation map. Then, by switching to the parallel coordinates these relationships can be examined more quantitatively. Let us now look at a few examples.

3.8.1 *Visualizing Clusters and Priorities*

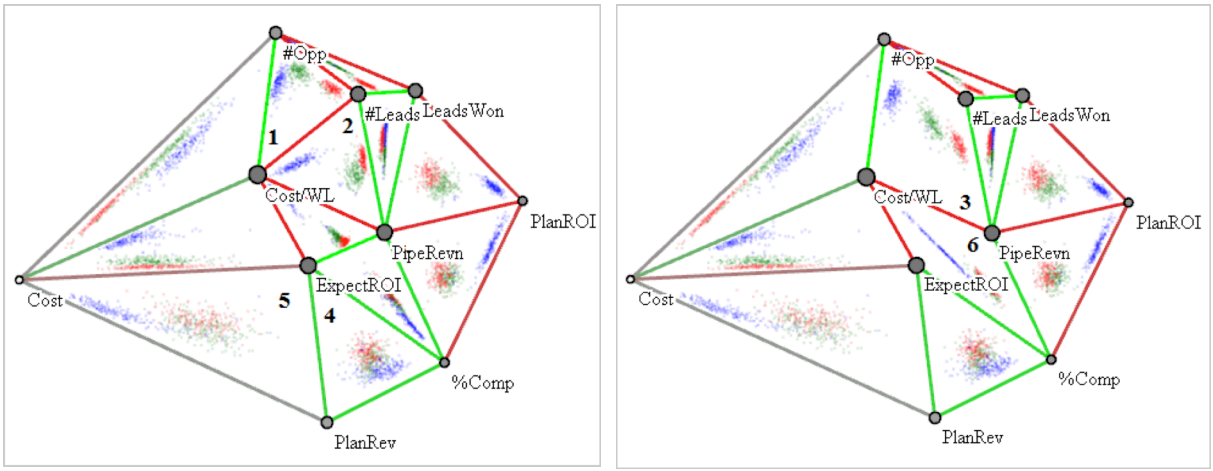


Fig. 3.16: Scatterplot analysis: (a) default layout, (b) mesh after removing the edge between subspace 1 and 3 as well as 4 and 5.

Fig. 3.16(a) shows the correlation map of the sales campaign dataset now augmented with the subspace scatterplots that were automatically generated by our tessellation algorithm. Already at first glance it becomes obvious that the blue team is quite different from the green and red teams—the latter two clusters overlap often while the blue cluster is disparate in most subspaces. In subspace 1 we observe that the red team's focus seems to be on generating many leads—but only few converted into opportunities. On the other hand, the blue and red teams seem to have better priorities by focusing on opportunities instead of leads. In subspace 2 we learn that the blue team spends much money per won lead, but this does not seem to translate to high pipeline revenue. In this aspect the red and green teams do better.

We now wish to get a more comprehensive picture about these issues and remove the edge separating these two subspaces. This generates subspace 3 in Fig. 3.16(b). Now we see the three teams well separated and learn that while the blue team does great on winning opportunities, it does poorest among all teams in terms of final pipeline revenue, while incurring more cost (relatively speaking) than the other two teams. Surprisingly, the red team (and also the green team to a lesser extent), despite the relatively few opportunities it creates, has a much better pipeline revenue emphasis than the blue team, possibly because it spends little money on its leads.

3.8.2 *Finding Appropriate Subspace Dimensionalities*

Our tessellated map effectively "unrolls" the high-dimensional space into the plane as a mesh of subspace scatterplots. But merging some of the plots can bring an even better understanding. In the previous example, we saw that for neither subspace 1 nor subspace 2 all three clusters could be separated at the same time. This indicates an insufficient number of dimensions for the subspaces the three clusters reside, and indeed we saw that by merging the two subspaces the clusters could be well separated. Hence, this configuration of the scatterplot mesh constitutes a better unrolling of the high-dimensional space, accounting for the intrinsic dimensionality of subspace 3. Similar is true for subspaces 4 and 5 in Fig. 3.16(a) where we see strong overlaps for some of the red and green cluster. Merging these two subspaces into subspace 6 (Fig. 3.16(b)) has a similar effect than with subspace 3—the three clusters are now much better separated.

3.9 CAUSATION ANALYSIS

Causation analysis is an analytical process that tries to answers the questions of causes, reasons, effects, results and consequences. It helps analysts make judgments or predictions by knowing the causes and effects. Here we are focusing on the problem of causation analysis from observational data.

Pearl [99] proposed to use causal networks, a type of graphical model, for causal reasoning. The fundamental idea is the use of a *Directed Acyclic Graph* (DAG) to represent direct causal relationships, as well as conditional-independence assumptions, between the system's variables. In the causal graph, each variable is represented as node. If variable A is a direct cause of B, then there is an edge going from A to B. In this case, we also say that A is one of B's parents, and B one of A's children. If there is a (causal) path from A to B, then A is an ancestor of B, and B is a descendant of A. However, how to build such a causal graph from multivariate, often purely observational, data has posed a great challenge for researchers because of the curse of dimensionality and the limited amount of data from controlled experiments, based on deliberate interventions. Pearl [99] formally states conditions under which we can learn causal networks from purely observational data. Unfortunately, those conditions are relatively limited and hard to prove in practice. Hence, it is up to the expert to decide whether a particular causal relation is valid or sensible or not. No statistical method can do that from observational data alone.

As we can see that our correlation network and Pearl's causal graph [99] share the same underlying concept: vertices correspond to variables and edges encode relationships. Although in many cases, the more robust the correlations are, the more likely they are to imply causation, using the correlation map directly for causal reasoning could be problematic. To extend our correlation network for causation analysis, we modify our correlation network by using a partial correlation based method for causal model discovery.

There are generally two types of causal discovery algorithms—score-based algorithms and constrained-based algorithms. The score-based algorithms focus on the robustness and implicit confidence measure that a likelihood-weighted combination of multiple models can bring; on the other hand, the constraint-based algorithms are able to handle data from any arbitrary potential causal graphs and turn it into clear, unambiguous causal output. We will concentrate on the

constraint-based algorithms in this section. The constraint-based algorithms search for conditional independencies $X \perp\!\!\!\perp Y \mid \boldsymbol{Z}$ in order to eliminate direct causal links X−Y from the *partial ancestral graph* (PAG). The representative algorithm is the Inductive Causation (IC) algorithm [100], which works as follows:

1. For each variable pair (X, Y) in the set of variables $\boldsymbol{V}$, look for a set $\boldsymbol{S}_{XY}$ such that X and Y are conditionally independent given $\boldsymbol{S}_{XY}$: ($X \perp\!\!\!\perp Y \mid \boldsymbol{S}_{XY}$); add an edge between X and Y if no such set can be found;
2. For each pair (X, Y) with a common neighbor Z, turn the triple into a V-structure $X \rightarrow Z \leftarrow Y$ if $Z \in \boldsymbol{S}_{XY}$;
3. Propagate the arrow orientation to preserve acyclicity without introducing new V-structures.

Because of the subset search in Step 1, the IC-algorithm has an exponential time complexity in the worst case, which makes it inappropriate to real-time interactive applications. To solve this problem, Pellet and Elisseeff [101] proposed an algorithm—Total Conditioning (TC) algorithm. Unlike starting from an empty graph in the IC-algorithm, the TC-algorithm works as follows:

1. For each pair (X, Y), add an edge $X-Y$ if the partial correlation $\rho_{XY \cdot V \setminus \{X, Y\}}$ does not vanish. We obtain the moral graph of G_0, i.e., an undirected copy of G_0 where all parents of the colliders are pairwise linked;
2. Remove spurious links between parents of colliders introduced in Step 1 and identify V-structures;
3. Propagate constraints to obtain maximally oriented graph.

Although when G_0 is a fully connected graph, the TC-algorithm has the same exponential complexity as the IC-algorithm, it could run rather fast in polynomial time in general since G_0 would have low connectivity in most cases. On the other hand, we could interactively reduce the connectivity by filtering out unimportant relationships. So we adopted the TC-algorithm in our framework to make it applicable to our interactive analytics scenarios.

Moreover, we apply the mass-spring model layout to the graph G_0 computed from the partial correlations in Step 1 of the TC-algorithm. The layout would place strongly partial correlated variables close to each other. In other words, highly conditionally dependent variables would be put close to each other. Consequently, these nearby variables would serve as a good starting point for further investigation of the underlying causal effects.

3.9.1 INTERACTIONS

We add various user interactions into the framework to reduce the computation complexity and to allow analysts to interactively modify the constraints and dependencies based on their expertise in the causal model discovery process.

First, after the mass-spring layout, highly conditionally dependent variables would be put close to each other. The analysts could simply use the mouse to select a cluster in the graph that they think are important to for further causal analysis, then the framework will run Step 2 and 3

for only the selected variables. As a result, the computation time could be significantly reduced by constraining the computation space—the selected variables.

In Step 2, the sequence of considering the edges influences the final causal model—the remaining causal links and even the causal directions. To let the analysts have control over the analytical process using their expertise, our framework let them interactively select any edge to start with; then the algorithm would run only for the selected edge to test if it can be removed to create the V-structure.

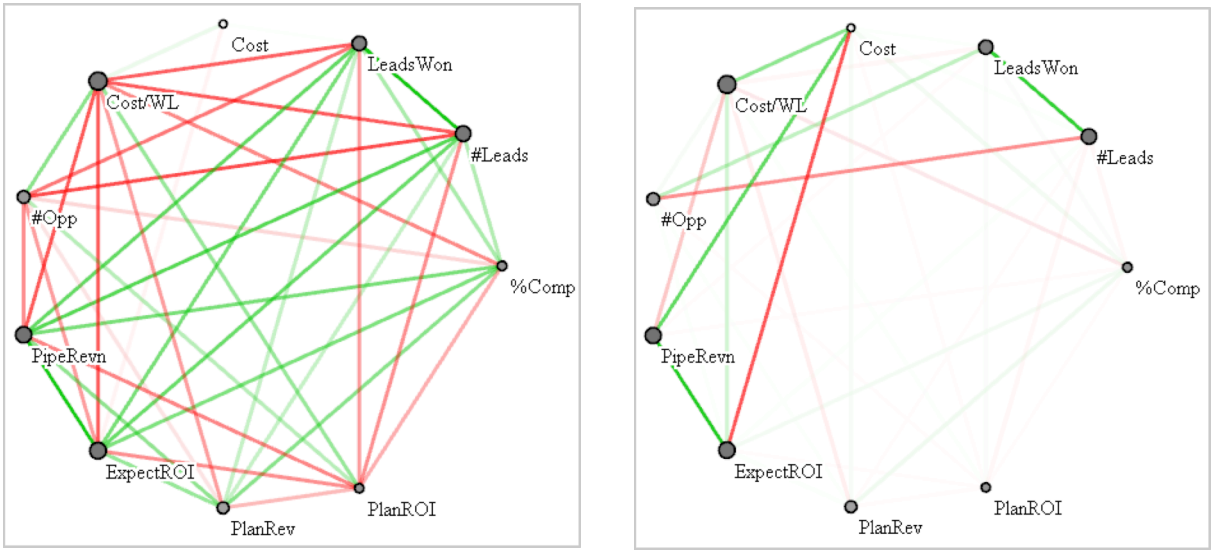


Fig. 3.17: Difference between (a) the correlation map and (b) the partial correlation graph G_0. The two graphs have not been laid out by the mass-spring layout for better comparison.

Moreover, sometimes the automated computation of Step 2 and 3 of the TC-algorithm would generate some false causal relationship. Our framework allows analysts to interactively set or modify any causal links in the graph. If the analysts set the causal link before Step 2, then we run Step 2 and 3 with the specified constraints—we will build the V-structure only if the direction of the edges in the V-structure is the same as the specified constraints. We also allow the analysts modify the causal direction after Step 3 if they think the causal relationship is wrong based on their expertise.

The framework also provides the analysts with several filters to filter out unimportant variables or relationship links. For example, the edge filter could be used to filter out the relationships that are not strong enough for further investigation. In other words, if two variables are not strongly dependent with each other, we could remove the relationship between the two variables before the time-consuming causal relationship computation.

3.9.2 CAUSAL ANALYSIS

We use the sales campaign dataset to demonstrate how to use our framework for visual causal analysis. First let us look at the difference between the correlation map and the initial graph G_0 built after Step 1 of the TC-algorithm computed from partial correlation, see Fig. 3.17.

We can see that compared to the correlation map, G_0 has a much lower connectivity. Thus the computation time for Step 2 and 3 would be greatly reduced. Then we could layout the variable via the mass-spring model to put highly dependent variable close to each other, see Fig. 3.18(a). We can see that there are clearly two main clusters within which the variables are highly partially correlated. From here, the analysts could either interactively select a region to go further investigation or run Step 2 and 3 of the TC-algorithm directly for the whole graph. In our example, the analyst used the edge filter to filter out some unimportant edges (relationships) to further reduce the computation space (Fig. 3.18(b)) and then run the algorithm for the whole

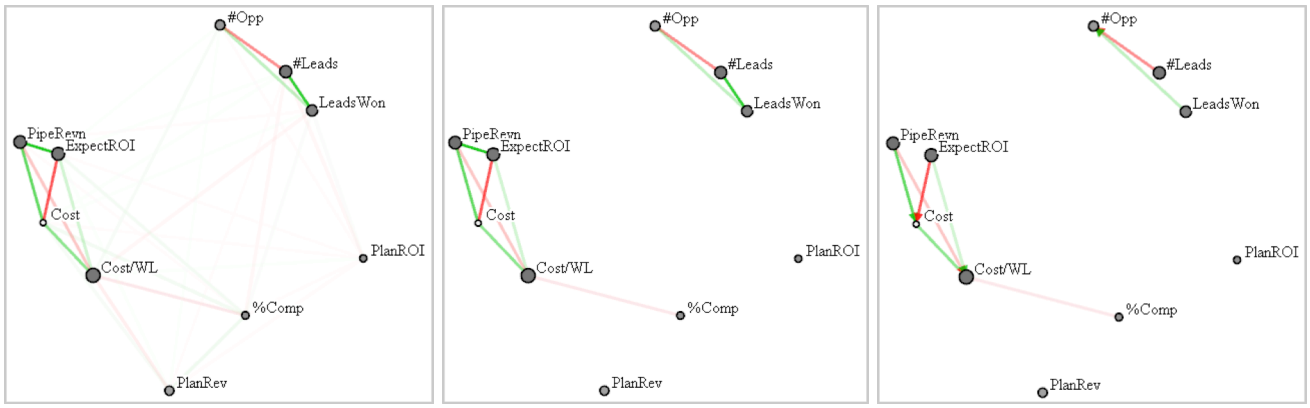


Fig. 3.18: Visual causal analysis. (a) Mass-spring layout of the initial graph G0. (b) Graph G0 with lower partially correlated edges filtered out. (c) The potential causal graph after Step 2 and 3 of the TC-algorithm.

graph. The result causal graph is shown in Fig. 3.18(c). We can clearly see that the number of opportunities (*#Opp*) is dependent on the number of leads (*#Leads*) and leads won (*LeadsWon*), while cost (*Cost*) is dependent on the pipeline revenue (*PipeRevn*) and expected return on investment (*ExpectROI*), which are consistent with our knowledge.

Admittedly, what our interface could do is just a small part of the whole causation analysis pipeline. There are much more topics in causation analysis. The contribution of our framework is that it provides the analysts with a useful interface to apply their domain expertise and to visually do causation analysis. The analysts could see the influences of their operations in real time, thus to make better further judgments.

3.10 Evaluation

We tested two aspects: the accuracy of the dimension ordering and the utility and effectiveness of the linked interface.

3.10.1 Dimension Ordering

We have compared our correlation-based TSP-ordering with other automatic ordering methods proposed in the literature, and the results are presented in Fig. 3.19. Here we used the synthetic dataset. The dataset has 25 dimensions ($A,B,C,\ldots,Y$) and 1,000 data points, and is made

purposely to have two subspaces – a 9D subspace (*A,C,E,F,G,H,R,U,V*) with two clear clusters and a 6D subspace (*D,I,J,L,M,P*) with three clear clusters (as shown in Fig. 3.19(d) and (e)). All others are noise dimensions. Fig. 3.19(a) shows the original ordering, and Fig. 3.19(b) uses the clutter-based ordering [99]. Though some of the structure can be observed (most noise dimensions tend to be next to each other and the presence of subspaces is apparent), the general structure is not clearly shown. Fig. 3.19(c) is using Ankerst's method [8]. The two subspaces are well captured, but the noise dimensions are split into two parts. The structure of the dataset is best represented in Fig. 3.19(d), which is generated by Ferdosi's SBF dimension ordering [38].

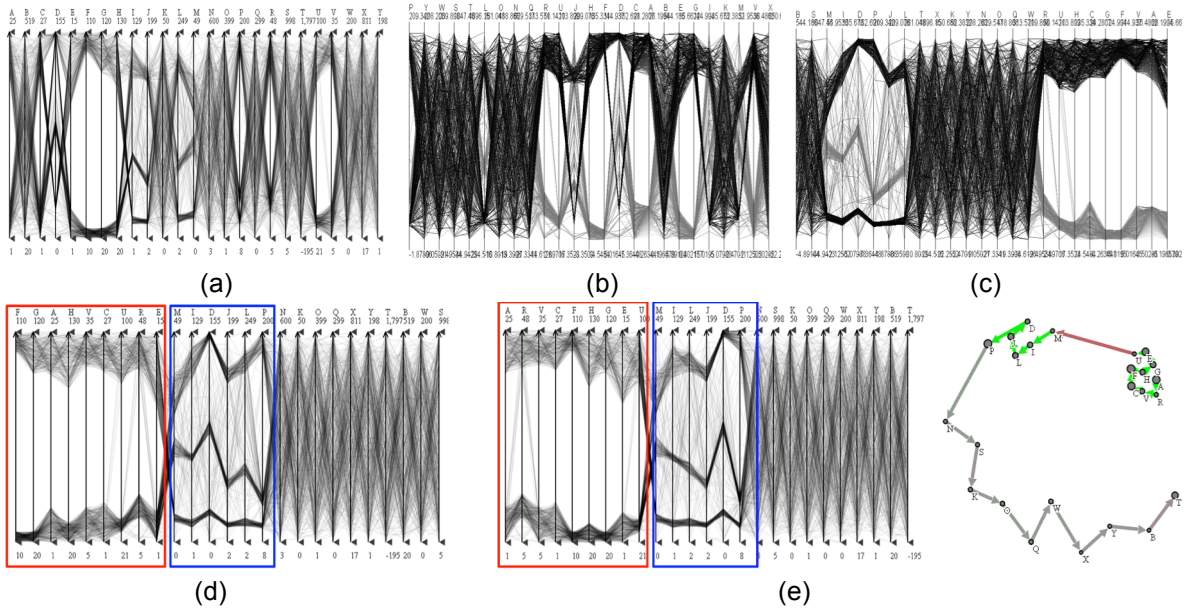


Fig. 3.19: Comparison of our automatic dimension ordering algorithm with other methods. (a) Original ordering. (b) The clutter-based ordering of Peng et al. [99] is unable to put the proper dimension order to show sub-clusters. (c) Ankerst's method [8] can capture the two subspaces, but the other un-related dimensions are split into two parts. Figures 8b and 8c are generated by xmdvTool [133]. (d) Ferdosi's subspace-based dimension ordering [38], which is able to capture the structure of the dataset (2 cluster subspace highlighted in red rectangle and 3 cluster subspace highlighted in blue rectangle). (e) Our method: the result is quite similar to (d).

The two subspaces and the noise dimension are well separated. Fig. 3.19(e) is obtained by our TSP-based approach. We observe in that the result is quite competitive with the SBF method, although the dimension orderings within the subspaces are different for each of the two methods. It appears that the TSP-based approach leads to dimension orderings in which the cluster values of adjacent dimensions change in a more linear fashion – compare the paths of the center cluster in the second subspace (blue window) in Fig. 3.19(d) and (e).

But despite these positive observations, we wish to note that the work presented here is mainly about our interactive network-based interface and not about the quality of the TSP-based dimension ordering. More comprehensive studies would be needed to validate the latter. But as the sales campaign example in Section 3.6 has shown, an automated ordering might not always

be useful and might require substantial edits to yield a possibly less optimal but more meaningful ordering, which our interface readily facilitates.

3.10.2 INTERFACE

We aimed for a framework that can even be accessible to users with no significant prior training in high-dimensional data analysis. So we performed a user study with members of this potential user group to get more insight into the effectiveness of our framework. We used the sales dataset because it contains some specific easy-to-grasp relationships. Our goal was to see whether and how quickly the test subjects could identify the relationships described in Section 3.6, i.e., why the blue team behaves differently from the others (using a fewer leads and won leads to generate more opportunities). Our hypotheses for the user study were:

H1. With the help of our network-based display, users are able to find the relationship more accurately.

H2. With the help of our network-based display, users are able to find the relationship faster.

To test these hypotheses we invited 18 graduate students (none majored in business) to participate. First we spent about 20 minutes to give them an introduction to our framework. We used the Cars dataset because this domain is the most generally familiar. We made sure that after this period all subjects knew the concepts of parallel coordinates and the network display and knew all the interactions supported by our framework. Then we randomly split the subjects into two equal-sized groups: one group only used the parallel coordinate display along with the raw data table (Group1), and the other group used both displays (Group2). We then asked each subject to select the attribute in the sales dataset that best explained the scenario elaborated on in Section 3.6.

In Group1, 3 students found the correct answer, i.e. *cost/wonLead.* In Group2, 7 students picked *cost/wonLead* because this attribute is the closest one with a dark red edge to *#leads* and *#leadsWon.* 1 student picked 3 attributes (*cost/wonLead, pipelineRev, and plannedROI)* which are nearby and said the scenario might be caused by the combination of them (regarded as 1/3 correct). So in this case we observed 7.33 (7+1/3) students with the right answer, more than twice than in Group1. Therefore our network display clearly helped. The corresponding *p*-value is 0.039, which means Hypothesis 1 is confirmed.

To test Hypothesis 2, we used an independent two-sample t-test based on equal sample sizes and equal variance. On average, participants spent more time to find the answers in Group1 (*Mean* = 20.22 seconds) than those in Group2 (*Mean* = 11.56 seconds). The corresponding *t*-value is 2.85 and *p*-value=0.018. For 18 participants (degree of freedom = 16), *t* must be at least 2.12 to reach $p < 0.05$, so this difference was statistically significant.

Also, among the 18 students, 11 of them claimed that it was the first time they had seen a parallel coordinate display. It was interesting to notice that these 11 students asked more questions and spent more time on learning the parallel coordinate system than on the network display. They stated that the network display was quite easy to understand since they had seen

similar displays before. Some mentioned that the network display reminded them of the "Get direction" feature in Google Maps. This insight suggests that our network-based navigation interface is quite accessible, even to novice users.

Chapter 4

GEOSPATIAL DATA ANALYSIS

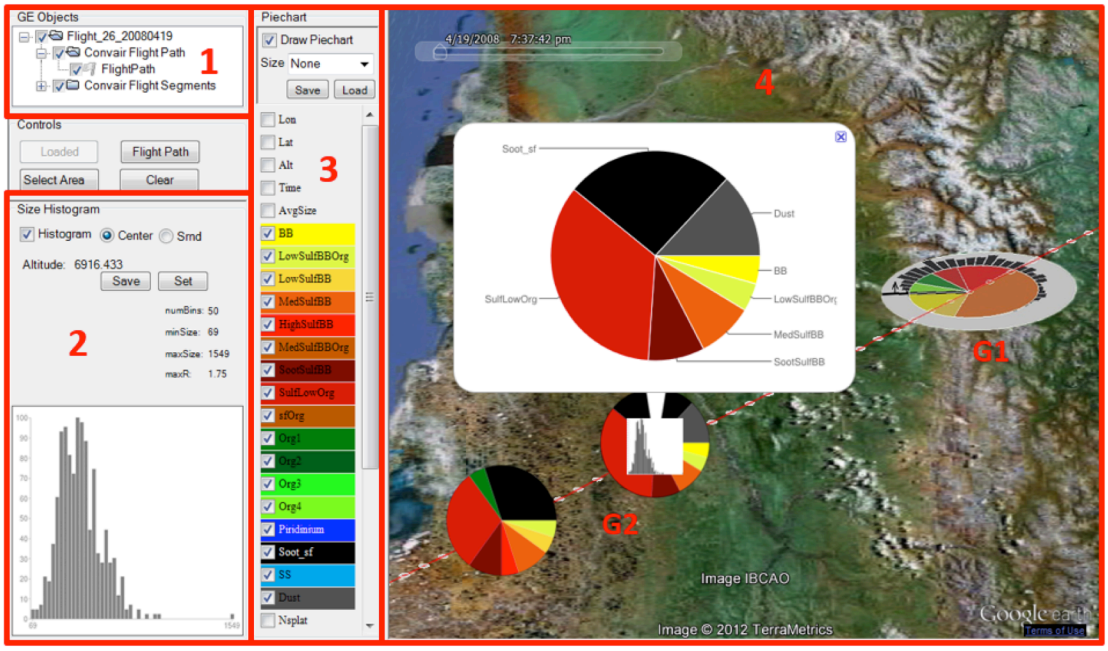


Fig. 4.1: Interface and pie chart-histogram design (*PHD*). Panel 1 is the GE Object control panel which allows users to show/hide a GE object. Panel 2 is the bar-chart (or histogram) panel – here configured for *ISDAC* particle size. Its control panel allows the analyst to control various parameters, including the design of the *PHD* in the GE display. Panel 3 is the pie chart control panel. It contains the parameters for configuring the pie chart, such as what attributes will be displayed in the chart and which attribute is used to determine the size of the *PHD*. Users can also save/load the previous settings or export the current pie chart/histogram information into text files for further research. Panel 4 is the Google Earth display showing the three different *PHDs* styles we provide. When the user clicks on the *PHD*, the pie chart detail is shown nearby, and the corresponding size histogram is shown in the histogram panel.

Climate research produces a wealth of multi-field data. Here, a multi-field is considered a multivariate extension of a scalar field, that is, each point in Euclidian space offers values of multiple properties at that location and time. The definition of field as a physical quantity associated with each space-time point is one that has been adopted by all branches of physics:

electricity, electro-magnetism, gravity, but also fluid dynamics and so on. Likewise, geography has adopted a similar notion of field than physics, but with the important distinction that geographic fields are not necessarily produced by strict physical laws. Rather, they can be due to demographic sampled assessments, behavioral modeling, population and cultural effects, environmental measurements, and many others.

These data often have a geospatial reference and so it is of interest to show them within their geospatial context. One can consider this configuration as a multi-field visualization problem, where the geo-space provides the expanse of the field. However, there is a limit on the amount of multivariate information that can be fit within a certain spatial location, and the use of linked multivariate information displays has previously been devised to bridge this gap. In this paper we focus on the interactions in the geographical display, present an implementation that uses Google Earth, and demonstrate it within a tightly linked parallel coordinates display. Several other visual representations, such as pie and bar charts are integrated into the Google Earth display and can be interactively manipulated. Further, we also demonstrate new brushing and visualization techniques for parallel coordinates, such as fixed-window brushing and correlation-enhanced display. We conceived our system with a team of climate researchers, who already made a few important discoveries using it. This demonstrates our system's great potential to enable scientific discoveries, possibly also in other domains where data have a geospatial reference.

Geographic fields are often visualized using cartographic techniques, contour plots, and choropleth maps. They are in many cases multivariate – just consider a map of households with incomes, number of children, cars, and so on. However, multiple variables are difficult to plot with choropleth maps, and so a number of researchers have linked them with multivariate visualization displays, most frequently parallel coordinates. In such a visual geo-analytical framework, the analyst would obtain insight about the spatial distribution of the color-coded populations in the geographical display, and then visualize the multivariate signatures/composition of these populations in the parallel coordinate display, using a color legend as a reference. Typically these geo-graphical displays are 2D maps.

The system in this chapter was an extension of the visual analytics framework that was described in the previous two chapters. It was developed in close collaboration with two groups of climate scientists. With the growing intensity of local climate fluctuations, the melting of polar ice caps and the emergence of other related processes, researching the cause of these trends has gained tremendous importance in recent years. As opposed to demographic data, phenomena that change the earth's atmosphere bear important 3D relationships, and as such climate researchers typically acquire their data using probes in 3D space, either aided by airplanes or other sensors situated at diverse altitudes. In their research, climate scientists often pursue efforts in which they seek to proof or disprove hypotheses involving different aspects of the data. This mandates an interactive system that not only supports both data types – spatial and the non-spatial fields – but also uses 3D maps for the geo-spatial display.

An attractive platform for interactive geo-visualization is Google Earth. It provides easy access to 3D geographical data and for that reason has often been lauded as a "democratization of GIS". In recent years, various efforts have also used Google Earth as a geographically-

referenced canvas for the presentation of many types of field data, both of physics and geographic nature. However, the purpose of these Google Earth-based data displays is typically merely data visualization, with only some selection capabilities available. The opportunities for user interaction are mostly restricted to navigating the environment – the globe – using the standard Google Earth spatial navigation tools and possibly a slider for time-animation of the data. To the best of our knowledge no application so far has utilized Google Earth as a platform for full-fledged visual analytics in which users can interact with the data directly in the Google Earth display, by ways of standard information interrogation techniques such as brushing, filtering, and aggregating, and communicating these interactions back to a linked information display.

This chapter demonstrates how our visual analytics framework would be helpful in climate research. Specifically, the major contributions of this chapter are:

1. We extend an interactive geo-browser – Google Earth – to support a set of common interactive information interrogation techniques such as brushing, filtering, and aggregating.
2. We link this extended geo-browser to a popular multivariate information display – parallel coordinates – and establish bi-directional interaction propagation.
3. We devise a new family of design primitives, conceived to show some multivariate data aspects directly in Google Earth.
4. We demonstrate the viability of our framework in the context of climate research, enabling a collaborating team of climate scientists to make important discoveries in their research domain.

4.1 MOTIVATION

A valuable element of our system is that its design is well-informed by the research workflow of climate scientists. This is confirmed by the fact that a team of such scientists was able to make a number of significant discoveries using our system (Section 4.4). While they might have made these discoveries with conventional tools as well, our system enabled them to make them much quicker and easier, and so accelerating progress in this important research domain.

Our efforts were motivated by two specific applications in climate science. In the following we describe their data as well as the research workflow for one of them.

4.1.1 DOMAIN DATA

ISDAC dataset

The ISDAC dataset was acquired by a single particle mass spectrometer (SPLAT II) [153][154][155] on Flight 26 (F26) which took place on April 19-20, 2008 as part of the Indirect and Semi-Direct Aerosol Campaign (ISDAC)[88], a month-long field campaign at the North Slope of Alaska (see Fig. 4.2(a), (b)). F26 began in Barrow, Alaska and ended in Fairbanks, Alaska (see Fig. 4.2(c), (d)). The flight began with a short transit over a DOE ground site,

(a) (b) (c) (d)

Fig. 4.2: Capturing the ISDAC dataset. (a) The Single Particle Mass Spectrometer (SPLAT II) operated by the collaborating scientist in-flight in the Arctic aboard a Convair-580 research aircraft. (b) Various sensor probes mounted on the aircraft wing. The aircraft flew various missions over Alaska to measure concentrations, size distributions, shape, density, and compositions of millions of particles in clear atmosphere to establish a large and highly resolved data set of Arctic aerosol particles. Other environmental variables, such as cloud density, pressure, and density were also sampled (c) Overview of the flight path and (d) profile as seen from the side, both captured with Google Earth.

followed by about 90 minutes of sampling a cloud at low-altitude. The aircraft then performed a spiral, climbing to an altitude of ~7000 m proceeding for a landing in Fairbanks. The main scientific objective of ISDAC was to improve the understanding of how changes in the size, composition, and concentration of aerosols particles influence cloud properties and their associated radiative forcing. During the month long campaign, SPLAT II measured the number concentrations, size distributions, shapes, densities, and compositions of millions of particles in clear atmosphere to establish a large and highly resolved data set of Arctic aerosol particles. In the cloud, SPLAT II characterized the properties of CCN particles, on which cloud droplets form, and those of interstitial particles to develop a highly detailed dataset.

The ISDAC dataset consists of more than 2 million data points, each a 33-D vector: latitude, longitude, altitude, time stamp, temperature, and pressure. It also contains measurements on the cloud particles (cloud droplets presence, cloud particle concentration, etc.) and on the aerosol particles (size and composition: soot, sulfate levels, organics, dust, sea salt, etc.). The dataset was obtained by fusing measurement files of different instruments using the time stamp for alignment/binning.

Global seawater oxygen-18 database

This Global seawater oxygen-18 database [118] is a collection of about 26,000 seawater measurements from all around the world, each an 8-D vector: longitude, latitude, month, year, depth, temperature, salinity, and oxygen composition ratio $\delta^{18}O$. The $\delta^{18}O$ value is a very good tracer of water origin and highly correlated with salinity, but it varies regionally and seasonally under some specific conditions. For example, when salinity is nearly 0 *psu*, then the $\delta^{18}O$ typically has a wide range of values. Possible reasons can be precipitation [31], river inflow [91], or glacier calving [39]. These various geographic and multivariate dependencies make this dataset an excellent test case for our system.

4.1.2 *Domain requirements for the ISDAC dataset*

The goal of the team of domain scientists we primarily collaborated with was to gain a better understanding of particle composition and size at various geo-spatial locations, as well as the relations to other particle properties, atmospheric conditions, and particle activation probabilities. Since the domain data share the calamities of most such datasets – many outliers, unspecified values for some attributes, etc. – these relationships are difficult to discern via automatic analytical algorithms, and this has motivated the use of visual analytics techniques to overcome these shortcomings [87].

In the following we list the set of basic requirements our collaborators expressed in the onset of the project:

R1. Ability to visually interact with the multi-field data.

R2. Ability to summarize the data in terms of different variables.

R3. Ability to visualize the relations among variables as a multivariate display

R4. Support of geo-spatial references, whereby the geo-spatial display should fully support interactions such as selection, filtering, and brushing.

R5. Support of coordinated displays – all displays should be linked such that operations on one display are reflected on the other.

Although our collaborators had access to a variety of visualization frameworks, such as parallel coordinates, our ClusterSculptor framework [94], Microsoft Excel, Google Maps and Earth, etc., these systems were disjoint and could not provide the holistic dual-domain interaction that was needed to produce the desired insights. For quite some time, the scientists would create pie charts of particle distributions with MS Excel and then overlay them on a static Google Earth map to visualize the data – clearly a rather cumbersome workflow which greatly slowed the pace of research.

4.2 System Design

Our primary aim was to devise an integrated framework that would allow climate scientists to interactively visualize and analyze their multi-field data. Here we had a number of choices. In the following we address each in the context of the five domain requirements (marked as Rx below) listed in Section 4.1.2.

4.2.1 *Visualizing multivariate data (R1)*

To visualize multivariate data, among the most popular techniques are parallel coordinates [60] and scatterplot matrices [53]. However, due to the distributed 2D tiled layout of the scatterplot matrix, it can be difficult to discern relationships that involve more than two variables. We chose parallel coordinates since they visualize high-dimensional data as flows across vertical

axes and so yield a more connected representation. They also conveniently support brushing, selection and filtering by simple axis interactions.

4.2.2 SUMMARIZING DIFFERENT VARIABLES (R2)

Pie charts (for proportions) and histograms or bar charts (for distributions) are fairly low-tech but well understood visualizations, and have also been widely used by our collaborators. Therefore, to reduce the learning curve we made use of these paradigms in our system, but merged them into a combined design for added expressiveness. We call this design the *Pie Chart-histogram Design* (*PHD*). In order to avoid potential overcrowding in the map, our primary goal was to make the *PHD* space-efficient. This ruled out designs that would place a pie chart next to a bar chart, as such configurations would waste much empty space. Conversely, as is well known [53], circles – when sized equally – achieve the tightest packing in 2D. Therefore, we strived to create designs with circular geometry. We derived two different designs mainly distinguished by their renditions of the histogram. Fig. 4.1 compares the two designs side by side, along with a pie chart with no histogram. One design leaves the pie chart non-occluded and wraps the bar chart around its perimeter. The other places the bar chart into the center but allows the pie chart sector lines to shine through. Finally, our interface also provides a dedicated window that shows the histogram of a selected space-point.

The feedback from our domain collaborators was that although the circular bar chart looked artistic, they felt that it was too different from traditional representations, and also difficult to read. They preferred the second rendition.

4.2.3 VISUALIZING RELATIONSHIPS (R3)

Here we chose an illustrative correlation rendering technique to display the correlation information in the parallel coordinate display. Since domain experts are not as fluent as visualization researchers in the visual language of parallel coordinate displays, we bridged this gap by adding illustrative hints to help the interpretation of trends [87] and correlations [150]. In parallel coordinates, negative correlations give rise to lines that aggregate into bow-tie shaped line bundles, however subtle. In [150] we proposed the following graphical-design inspired scheme that can make these relationships more obvious for less experienced users. First, for each adjacent dimension pair a bounding hull of the line bundles is computed based on the dimensional means and standard deviations. If the correlation is positive, then we can use this bounding hull as an abstracted band shape. Conversely, if the two dimensions are negatively correlated, the characteristic bow-tie shape is employed. Then the bounding hull is colored in terms of correlation strength where less saturation maps to lower correlation. Fig. 4.3(e) provides an example for this scheme.

4.2.4 SUPPORTING GEO-SPATIAL REFERENCES (R4)

Map-based methods, such as Google Maps, Bing Maps, and Google Earth, have been widely used to provide geo-spatial or location references in many applications. In climate research, the

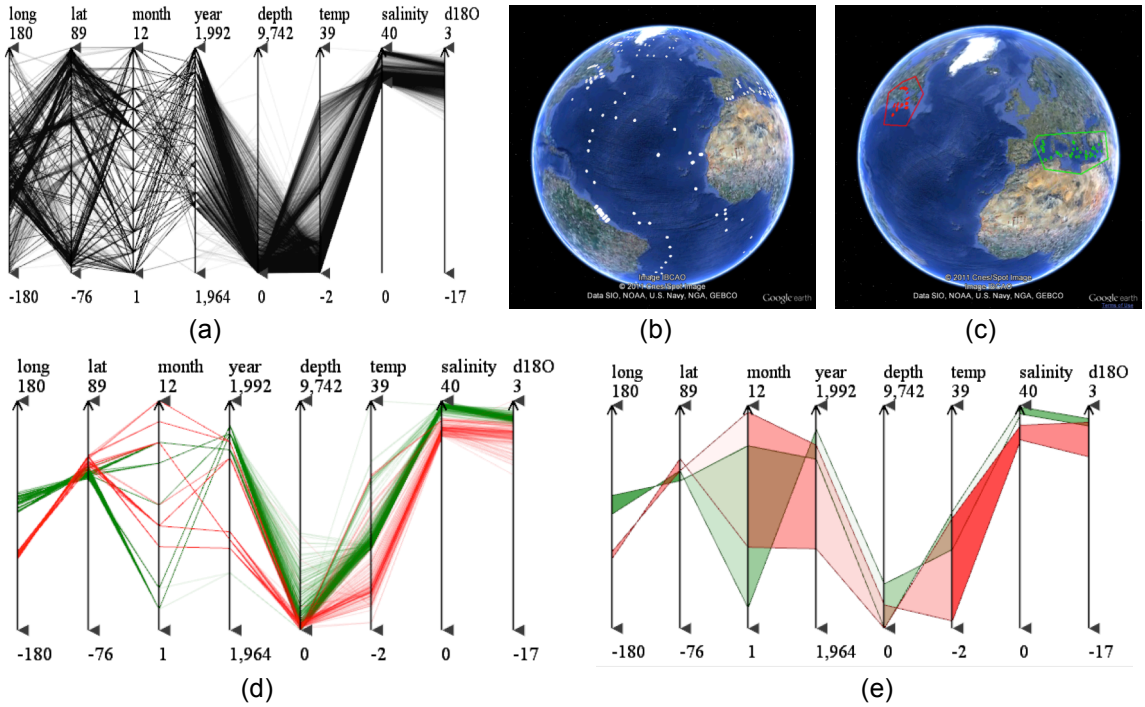


Fig. 4.3: Dual-domain analytics. (a) The analyst first uses the PCP brushing handles to select the normal ocean data points (salinity from 32 to 40). (b) The GE display responds by showing only these remaining data points. (c) Next the analyst uses mouse clicks to outline some interesting regions in the GE display (Mediterranean shown in green and Gulf of St Lawrence shown in red). (d) The points inside the selection polygon appear highlighted in the PCP display. (e) Correlation-enhanced PCP display.

altitude (elevation or pressure) plays an important role in the analytical process. Although 2D map-based methods can show the entire world in one display using Mercator projection and the like, due to the fact that the altitude information will be inevitably lost after projecting the 3D data onto the 2D maps, they are not useful for our purposes. Hence, we chose Google Earth as the geo-spatial reference display.

4.2.5 *Supporting coordinated displays (R5)*

Our framework consists of two displays: a multivariate visualization display and a geographic display. These two displays are linked such that operations on either display will be reflected on the other. Upon reading a dataset, longitude, latitude, and altitude are used to populate Google Earth (GE) with simple icons (placemarks). Meanwhile, the parallel coordinate plot (PCP) is populated with the data spectrum of the data points. Analysts can then use the mouse and brush in either display (GE or PCP) to select a subset of points, assign a color to these points, and see them reflected in the same color in the other display. To make up for the shortcomings of GE to display multivariate information, we have stricken a compromise and display a pie chart and a histogram at each selected measurement site (see Section 4.2.3). Both

size and color of the GE site icons can be linked to any variable using the attribute mapping checkboxes in the pie chart control panel.

4.3 METHODS

Koua et al. [75] proposed 10 general exploratory goals that geo-analytic systems should address: identify, locate, distinguish, categorize, cluster, distribute, rank, compare, associate, and correlate (see Section 4.4.2 for a more detailed description in the context of our system). Since geo-data have two types of attributes – geospatial and non-geospatial – only an interface that links (at a minimum) two dedicated displays – one geospatial and one multivariate – together can achieve a comprehensive visualization experience that meets these goals. This was recognized already early on. However, another issue is how users can *express* their goals, which they typically do via manual *selection* operations – one might call them gestures. A PCP information display typically facilitates selection interactions by allowing users to manipulate range handles on the individual axes. On the other hand, a geographical display such as GE would support selection operations by allowing users to click on points (or placemarks) or draw bounding contours around sets of points. To the best of our knowledge these direct selection interactions are thus far not supported in GE-based visual analytics systems.

Finally, once these data points have been selected in GE and color-tagged they would be represented in the PCP display as a group in the same color (and vice versa). This completes the full *brushing* operation.

Our framework is developed in C# using Direct3D for graphics. Our GE display uses the GE plug-in and also a C# custom-built API for the GE plug-in – the *Winforms-Geplugin-Control-library (WGCL)*[21]. This library defines a list of methods that can be used to interact with the GE plug-in by dynamically injecting JavaScript code into a browser page during run-time to interact with the GE plug-in. WGCL works in the .NET Framework (C#) and provides a bridge between the client-based application and the web-based GE plug-in.

4.3.1 EMBEDDING THE PHDS INTO GOOGLE EARTH

All embedded pie chart histogram designs (*PHD*) are rendered on the fly whenever a data point is selected. Fig. 4.4 describes this process as a flow chart. On the other hand, if a region is selected by brushing either the GE or PCP display, the corresponding *PHD* for all selected points will be shown and placed nearby the average geographical location.

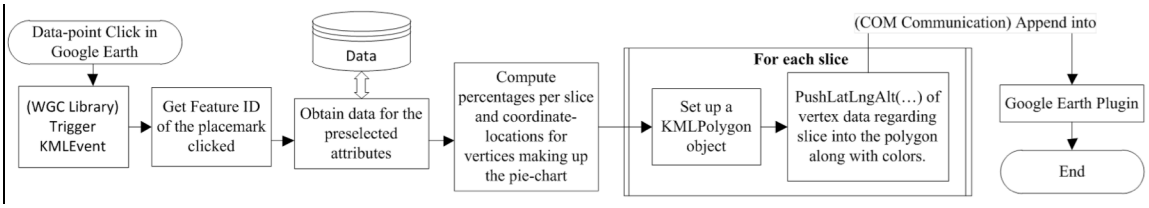


Fig. 4.4: Flowchart showing how to dynamically render an object into Google Earth. Here the pie chart is used as an example.

There are in fact two approaches to render a *PHD* in GE. One is to render each *PHD* using GE's polygon rendering functionalities (see G1 in Fig. 4.1). A downside of this approach is that after rendering, both location and tilt angle of the *PHD* are fixed in GE. This means that if we wish to make the *PHD* always face the viewer, we must delete it from GE and re-render it every time we change the viewing direction, which can be rather time consuming. However, this method provides better depth and height information than the second method in which we use an icon (image) to represent the *PHD*. GE supports a feature in which an icon/image of a placemark consistently faces the viewer, no matter how we change the view direction. But the challenge here is how to render the *PHD* image on the fly. Fortunately, Google Chart Tools (https://developers.google.com/chart/) provide good support for rendering various charts into an image which can be later retrieved via a URL. Using Google Chart Tools, we render the image of a *PHD* by passing its parameters (such as pie chart compositions and histogram information) and then load the image into GE by passing it the image URL. The result is shown in G2 in Fig. 4.1. In our system, we allow the user to choose the rendering method, dependent on the underlying task.

4.3.2 BRUSHING IN THE GOOGLE EARTH DISPLAY

We support two types of brushing tasks in GE:

Single data point brushing. The user can select any data point in GE by a single mouse click. Attributes associated with the data point will be shown in a popup window and at the same time the corresponding polyline will be highlighted in the PCP display.

Region-based brushing. This addresses the need to visualize the behavior of an entire geographic region in the PCP display. However, since mouse dragging is reserved by GE for view rotation, we cannot simply drag the mouse to outline the region of interest. Instead, we impose the moderate requirement that users can use a series of mouse clicks to specify the vertices of the selection polygon. The polygon can be either convex or concave. After polygon completion, we employ a quick points-inside-the polygon test to determine the data points inside the selection polygon. These interior points are then marked and the corresponding polylines highlighted in the PCP display. Likewise, selecting points in the PCP display will highlight them in GE. Fig. 4.3 shows an example for both brushing directions.

4.3.3 BRUSHING IN THE PCP DISPLAY

In our PCP display users are able to manually interchange axes, flip (invert) axis directions, and perform statistics-guided outlier filtering [62] and clustering in each dimension. The following are a set of further capabilities our domain scientists found useful.

Fixed-window brushing. We extended the range handles typically used to bracket data intervals from the top and bottom to *fixed-window brushing* – essentially an interval slider. In this mode, the distance between the two handles remains fixed and as the user drags the handles they will move up and down simultaneously. This feature is very helpful to show how other

attributes behave as one attribute changes, and the GE display will visualize the corresponding changes in geo-spatial locations.

Adjust-window brushing. Users can also drag the mouse up/down to make the width of the bracketed window bigger/smaller. This operation is useful to see the behavior of the entire dataset as the range of an attribute spreads out/shrinks.

4.3.4 ADDRESSING EXPLORATORY TASKS

To rate how well our system supports geo-analytics we examine our framework via Koua's exploratory tasks, also in the context of our domain applications.

Identify. The now fully bi-directionally linked displays make it easy to identify relationships.

Locate. GE provides an effective way to directly see not only the longitude and latitude information, but also the altitude information, which is a significant factor in climate research.

Distinguish and distribute. By using the *PHDs* in GE, users can easily judge the differences among sample points directly in the geographic domain.

Categorize and cluster. The range-handle, cluster, and tag operators in PCP and the point/region selection operators in GE well support interactive classification directly in the most suitable domain. Any cluster can be subdivided further or re-assigned to another cluster, all of which is maintained via a cluster checkbox.

Rank and compare. The *PHDs* in GE provide the user with an effective means to visually assess ranking and perform comparisons. Likewise, the brushing and clustering operations in PCP allow users to compare cluster behaviors in a multivariate context.

Associate and correlate. Both *PHDs* and brushing operations in PCP and GE aid users in assessing relationship between attribute(s) and geographical information. In addition, the PCP correlation visualization allows users to easily recognize data relationships.

4.4 USE CASES AND RESULTS

We initially tested the various features of our system with a dataset from ocean modeling, and then developed our framework further in collaboration with our team of climate researchers. We describe both applications next.

4.4.1 GLOBAL SEAWATER OXYGEN-18 DATABASE

First, since the data is an amalgamation from different sources and different tasks, several attributes have undefined values (-999 for depth and -100 for temperature, salinity and $\delta^{18}O$ in Fig. 4.5(a)). These undefined values can significantly influence the analysis results. As shown in Fig. 4.5(b) very weak correlations among all dimensions are observed. Following, Fig. 4.5(c) shows the parallel coordinate display after brushing the four dimensions, filtering out the undefined values and renormalizing their bottom axis brackets. We can now readily see in the

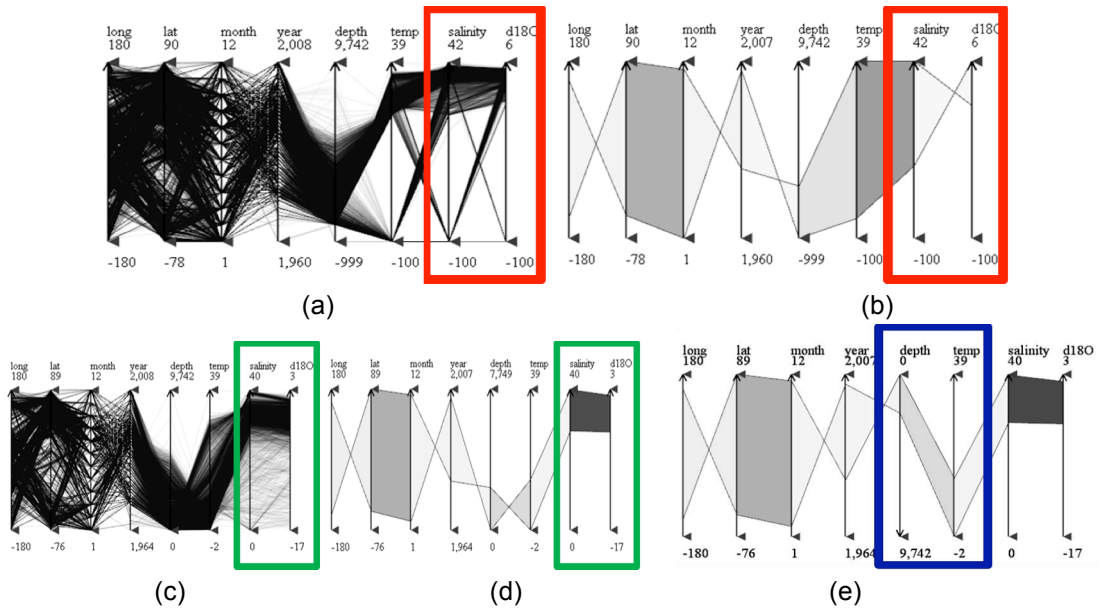
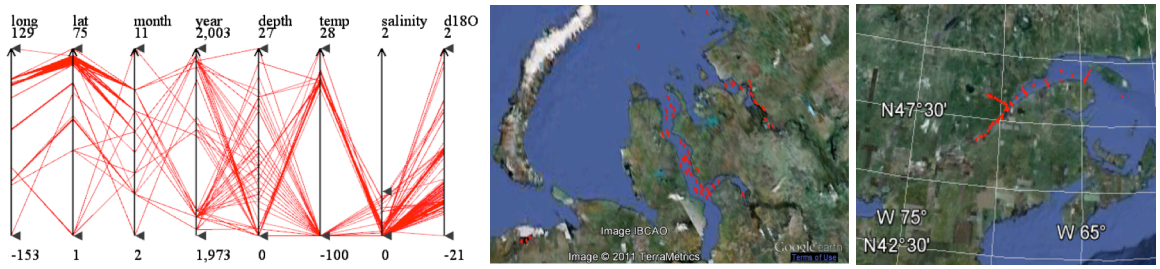


Fig. 4.5: Interactions in the PCP display. (a) PCP and (b) correlation-enhanced PCP display of the original dataset with undefined values in dimensions *depth, temp, salinity* and $\delta^{18}O$. (c) and (d) The same set of displays after filtering out the undefined values and zooming into the dimensions. We now have a clearer view of the remaining data. For example, we can see a strong positive correlation between *salinity* and $\delta^{18}O$ (green square in (c) and (d)) after the dimension zoom-in, while in the original dataset with undefined values, $\delta^{18}O$ is falsely negative correlated with *salinity* (red square in (b)). (e) Correlation display after axis depth inversion. Sea *Depth* and *Temperature* now have a positive relationship (blue square in (e)), which is consistent with our knowledge.

correlation display (Fig. 4.5(d)) that there is in fact a strong correlation between dimension salinity and $\delta^{18}O$.

Though salinity and $\delta^{18}O$ are highly correlated with each other, they behave quite differently at some conditions. One condition is when salinity is nearly 0 *psu*, the $\delta^{18}O$ might have widely differing values. Factors that can influence the $\delta^{18}O$ values are precipitation [31], river inflow [91], or glacier calving [39]. To get insight on which factor influences the sample points most, we first use the PCP brush handles to select the region that have nearly 0 *salinity*. We notice that the $\delta^{18}O$ varies greatly (see Fig. 4.6(a)). To determine the reason, we turn to the GE display (Fig. 4.6(b), (c)). After zooming into all the regions that contain the filtered data points, we can clearly see that all the remaining points are at the mouths of rivers, that is, where rivers meet the sea. Thus for this dataset, the most influencing factor appear to be junctures where fresh water meets salty water.

Our framework can also be used to confirm (or reject) hypotheses. For instance one hypothesis for sea water oxygen is that sensitivity of $\delta^{18}O$ can be greater than that of salinity in the deep ocean [16]. To test this hypothesis, we first use the brush handles to select the deep ocean dataset, here we choose depth > 2000 (Fig. 4.7(a)). From the poly lines, we observe that the $\delta^{18}O$ varies more than salinity does, but we cannot see the difference clearly. To get a better



(a) (b) (c)

Fig. 4.6: *Salinity* is nearly 0 psu while the $\delta^{18}O$ has widely differing values due to river inflow. (a) Brushing in PCP to select data that have near-zero salinity. These areas are: (b) Obskaya Gulf (estuary of Ob River), Yenisey (estuary of Mal. Taz River) and White Sea in Russia. (c) St. Lawrence River area in Canada.

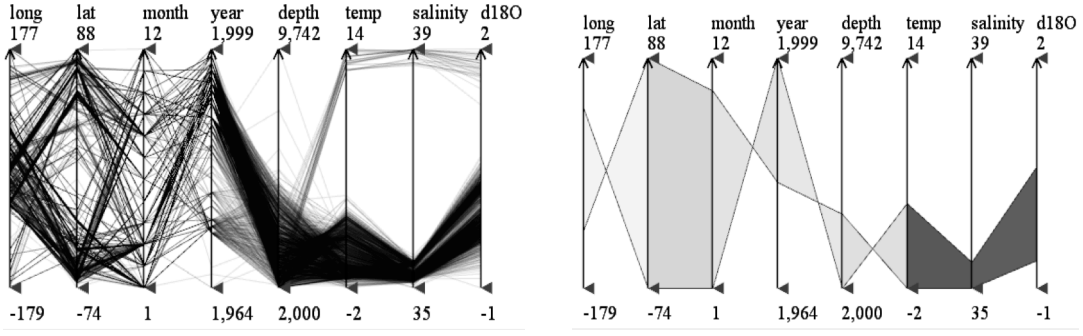
view, we set the PCP rendering mode to correlation-enhanced (Fig. 4.7(b)). We observe that the width of the branches for $\delta^{18}O$ is larger than that for salinity. Since each dimension is normalized in the PCP display, the hypothesis is confirmed.

The dual-domain analytics facilitated by the two linked displays also helps to make some interesting discoveries. For example in Fig. 4.3, the analyst first used the brush handles to select the normal ocean data points (salinity from 32 to 40), shown in Fig. 4.3(a). The GE display then only displayed the data points that were sampled in normal ocean waters. The analyst then outlines some interesting regions in the GE display – the Mediterranean area (green cluster in Fig. 4.3(c)) and Gulf of St. Lawrence (red cluster). Following these interactions, the points inside these selection polygons appear highlighted in the PCP display (Fig. 4.3(d)). By comparing these two clusters in the PCP display we can make the following observations:

(1) The samples observed in the Gulf of St. Lawrence appear much earlier (year) than those in the Mediterranean.

(2) The depths of the sample points in the Mediterranean have higher variation than in the Gulf of St. Lawrence.

(3) But the temperature, salinity and $\delta^{18}O$, the values in the Mediterranean have a much



(a) (b)

Fig. 4.7: Deep seawater (*depth*>2000) analysis. Both displays show that at deep sea the variation of $\delta^{18}O$ is larger than salinity.

lower-variant distribution than those in the Gulf of St. Lawrence, which means the water conditions in Mediterranean are more stable.

The observations can be confirmed in the correlation display (Fig. 4.3(e)) and are actually easier to see.

4.4.2 ISDAC DATASET

Much of the visualization work conceived for this paper was developed in tight collaboration with a team of climate researchers studying the effects of aerosols on global warming. One might consider the following Section a result of a formative, user-in-the development-loop user study.

Background

Atmospheric aerosols play an important role, affecting both global and regional climate change during the last century. They do so by scattering and absorbing solar radiation and by determining cloud properties [4][84] in their role as cloud condensation nuclei (*CCN*) and ice nuclei (*IN*). Yet the relationship between the properties of aerosol particles and clouds, i.e. the aerosol indirect effect, remains the most uncertain aspect in our current understanding of climate change. Scattering and absorption probabilities depend on particle number concentrations, size distributions, individual particle compositions, and on relative humidity (*RH*), which requires sophisticated instrumentation and data analysis tools to mine the vast amount of detailed data that needs to be acquired.

Laboratory data suggest that cloud formation and cloud properties are tightly connected with the properties of the particles, on which they form. Because particle's hygroscopicity, which is determined by particle composition, is related to their *CCN* activity, we expect particle compositions to play an important role in determining cloud formation and properties. The fraction of aerosol particles that activate to form cloud droplets depends on particle composition, size distribution, and number concentration, presenting a complex dependence. Most importantly, because *CCN* activation is not linear, it is essential to know the spatial distributions of the aerosol fields with high resolution.

The Arctic region represents an important and interesting location to study the forces that affect the global climate. Arctic aerosols are advected into the region from Asia, Europe, and North America, their loadings, compositions, and other properties vary significantly with meteorology. Biomass burning aerosol (*BB*) transported from Asia and North America is presently one of the most significant aerosol sources in the Arctic Spring. During Arctic spring, high *BB* concentrations produce the "Arctic haze". One of the interesting aspects of Arctic haze is that it is often found to be in distinct stratified layers [13]. Previous measurements of aerosol chemical composition point to sulfates as dominant constituents of arctic aerosol, with smaller contributions of soot, sea salt (*SS*), organics, and dust [51][108][109][124]. However, these composition measurements provide information on the bulk aerosol composition only with poor spatial and temporal resolution.

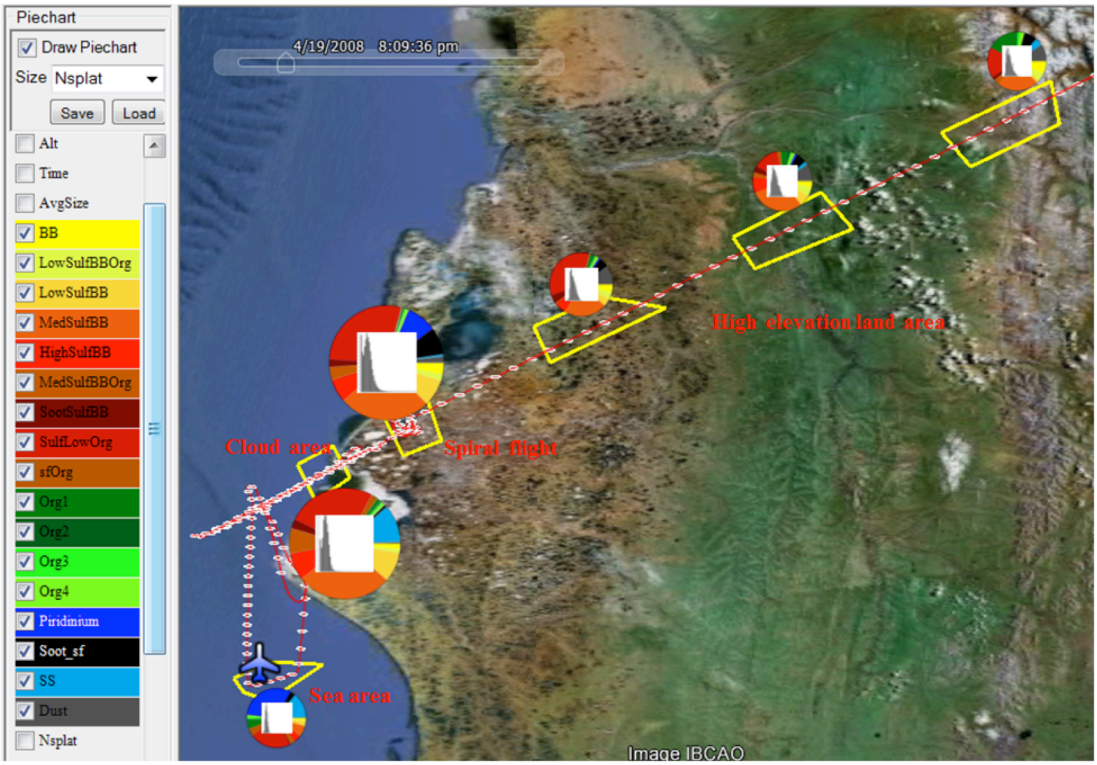


Fig. 4.8: An overview of particle compositions and size changes along the flight. The flight track is marked as a red line and each of the one-minute-spaced data points is superimposed as a grey ellipse. The polygon selection tool is used to outline several interesting areas (indicated by yellow polygons and labeled in red) and the corresponding *PHDs* are drawn nearby. *PHDs* are sized by the variable *NSplat* to allow an assessment of compression. The colors for the pie charts are assigned by the scientists via a color map popup widget, applying their domain standards. This overview visualization shows significant spatial variability in particle composition and size, which is consistent with previous reports that show a highly stratified atmosphere.

The effects of aerosol in climate models strongly relate to size and composition of individual aerosol particles [89], requiring high sensitivity and temporal resolution aircraft measurements. Because these types of measurements generate massive amounts of complex, multidimensional data there is a defined need for a specialized data analysis tool. In the following, we illustrate how our collaborating climate researchers, who are co-authors of this paper, used our framework to analyze their ISDAC dataset (see Section 4.1.1). Prior to this analysis, we first combined classes with particles of similar compositions to 17 distinct classes.

Visual analytics outcome

As a first step, to get an overview of the particle compositions, sizes, and compression information, we use the region brushing feature in GE display to outline some interesting regions during the flight, such as the low elevation sea area, cloud area, spiral area, and high elevation land area (see Fig. 4.8). Then for each region, the system computed the *PHDs* based on the summation of the particle compositions and particle sizes and plotted it nearby. The *PHDs* are sized by the *NSplat* variable to visualize compression information. The overview shows

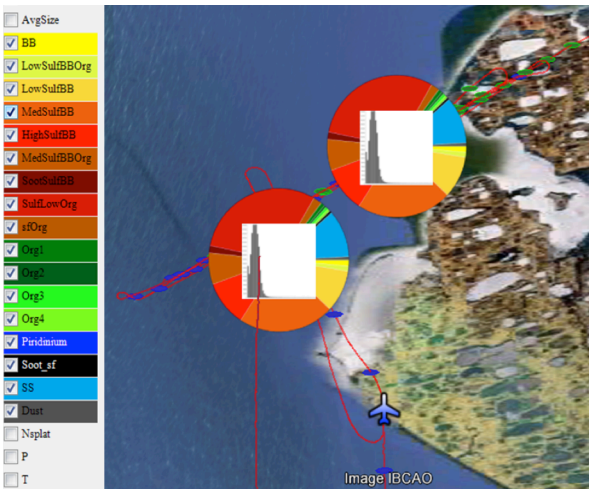


Fig. 4.9: Cloud data analysis. Here the scientists first used the PCP display to filter data points with clouds (*Nd*>20). They then applied a second filter to select only data points in which cloud droplets are sampled (*CVI* =1, green points on the flight track) to make the upper *PHD*. They then changed the filter to select data points where only the particles between the cloud droplets are sampled (CVI=0, blue track points) and obtain the lower *PHD*. The fact that these two *PHDs* are virtually identical is a significant discovery in climate research. This has never been done before and changes much of what has been assumed in the field.

significant spatial variability in particle composition and size, which is consistent with previous reports showing a highly stratified atmosphere. Most are *BB* (Biomass Burning) particles that were transported over long distances during which they adsorbed sulfates and additional organics. As the aircraft approached Fairbanks, at high altitude, the number of organic, dust and sea salt particles increased.

Next they used the framework to perform a visual analysis for the cloud particles (Fig. 4.9). Here they used the PCP to filter data points within the clouds (*Nd*>20). They then applied a second filter to select only data points in which cloud droplets (*CVI* =1, blue points) were present, creating the upper *PHD* in Fig 4.9. Next, they changed the filter to select data points where only particles between the cloud droplets (*CVI*=0, green points) were present, giving rise to the lower *PHD* in Fig. 4.9. A simple comparison between the two pie charts indicates that the compositions of activated and un-activated, interstitial particles are virtually the same. According to our collaborators, this is a significant discovery and changes much what has been assumed in the field so far.

The feedback from our collaborator was tremendously inspiring. They first checked that the analysis of clouds probed on all other flights yielded the same results. This provides, for the first time, direct experimental evidence that particles compositions play only a minor role in determining cloud activation probability. This finding is in contradiction with laboratory experiments that have consistently shown a simple relationship between activation probability and particle hygroscopicity, which is directly related to particle composition. It suggests that laboratory based cloud activation instruments operate on a different principle than that controlling cloud activation in the ambient atmosphere. More importantly, climate models use laboratory derived composition dependent activation probabilities to simulate cloud formation. The data presented here, if reproducible in other parts of the globe, indicate that cloud activation needs to be reformulated.

As noted above, stratification is one of the more interesting features of the Arctic atmosphere. To be properly characterized it requires instrumentation with high temporal resolution and data analysis tools that make possible to mine the large and complex data these instruments produce, with the appropriate resolution. In Fig. 4.10 we present an example in which we analyze changes

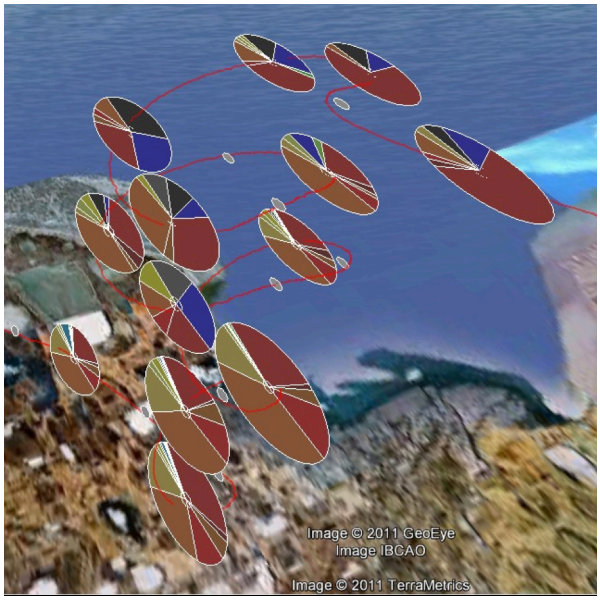


Fig. 4.10: Changes in particle composition as a function of altitude. We zoom into the flight's spiral ascent, where the aircraft climbed from a few hundred meters to an altitude of about 7000 meters. The pie charts clearly illustrate that particle compositions change significantly with altitude and that the changes are not monotonic. Here we use the first *PHD* rendering method (G1) in Secion 4.3 because it better shows the alleviation.

in particle composition as a function of altitude. We zoom in on the spiral ascent, where the aircraft climbed from a few hundred meters to an altitude of 7000 meters. Our program makes it possible to view the particle composition and particle size distributions at each data point by simply clicking on its icon. The figure shows 13 pie charts, clearly illustrating that particle compositions change significantly with altitude and that the changes are not monotonic. Similar filamentous structures are observed on horizontal legs as well (not shown here) and exhibit large changes in include large changes in particle number concentrations and size distributions as well. Our collaborators state that these complex structures have very important implications for climate modeling as well.

Chapter 5

HEALTHCARE ANALYSIS

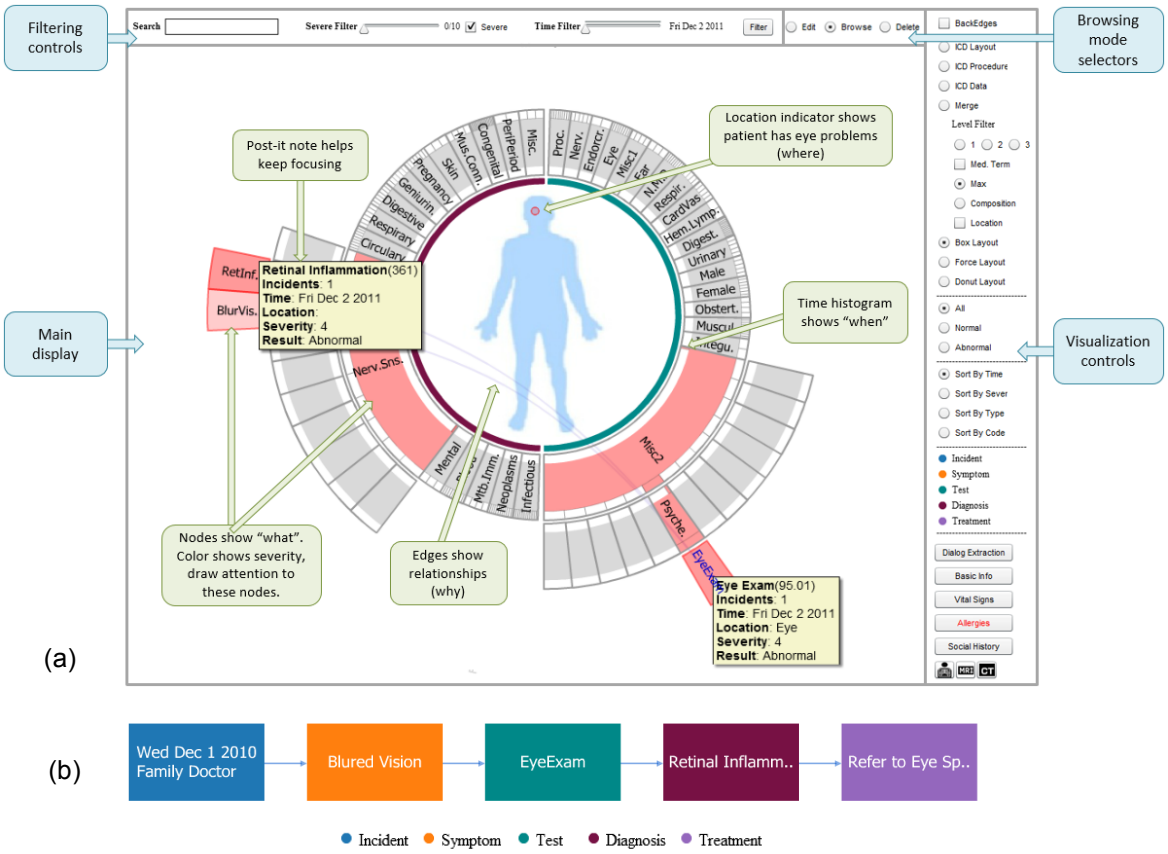


Fig. 5.1: The two coordinated displays of our system. (a) The radial (patient overview) display with integrated body map, along with the user interface. (b) The corresponding sequential (diagnostic reasoning) display using the same color coding. The user interface is identical to the radial display but removed here to save space.

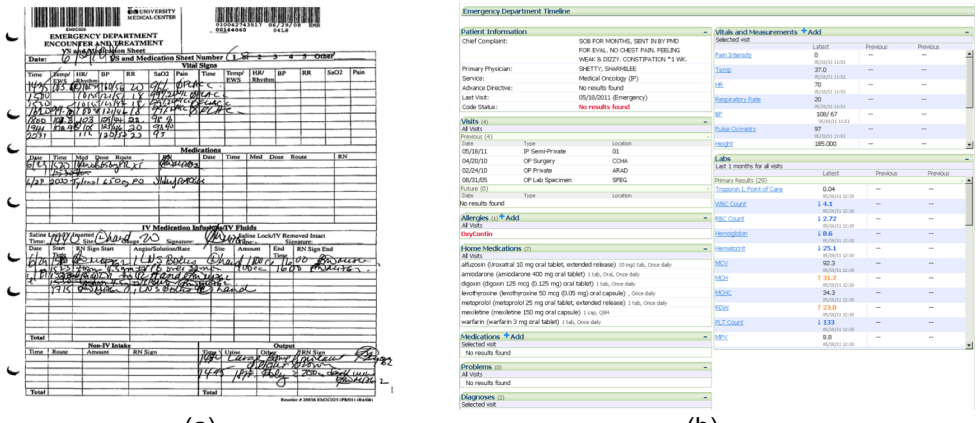
A central task of visualizing the multivariate data is to find the appropriate visualization paradigm for both the data and the problem scenario at hand. Many such visual information mappings exist [55], but it is well understood that there is no one method that can encode all

aspects of a given scenario, once sufficiently complex, and so the concept of multiple coordinated views has become an established paradigm [97]. Fluid interaction among these views via cross-filtering [134] and brushing [25] is the key to successful information (and data) exploration. Providing overview and detail-on-demand [121] is equally important—salient information should become available on a whim when requested but just as quickly disappear when no longer relevant. The interface we propose adheres to these well-established eminent requirements.

To structure the information domain and provide a suitable visual mapping for each we utilize the *Five W's* (*who*, *when*, *what*, *where*, and *why*) of journalistic reporting.

The Five W's are the elements of information needed to get a full story. They are encountered in many domains, such as a police detective investigating a crime or a market analyst planning an effective marketing campaign. The order in which the information is gathered or interrogated can vary case by case—crucial is only that all five W's are ultimately addressed. We believe that this grounding also fosters the new effort of storytelling in information visualization [115]—it will ensure that all aspects are covered in this visual story.

Our work demonstrates the application of the Five W's to health informatics. We find that most current healthcare informatics systems, if not all, lack the mantra of information visualization—overview and detail-on-demand—making it difficult to get a quick and effective assessment of a patient's state of health. Information is poorly organized and hard to obtain, and this has been blamed as the prime reason for the slower than expected adoption of the Electronic Medical Record [163]. It applies to both acute clinical encounters in emergency room scenarios [151] as well as to doctor-collaborative diagnosis and treatment plan development. Progress has been made in terms of temporal patient-centric organization and other statistical dimensions [3][104] [147][148][149], but these have rarely been linked and comprehensively organized. We propose to use the Five W's as a means to establish a comprehensive multi-faceted assessment of the patient and his/her history. We then associate each such W with a dedicated, linked visual encoding that can represent and communicate it to the other W's in effective ways.



(a) (b)

Fig. 5.2: Evolution of the medical record. (a) Paper-based. (b) Electronic. Images were shrunk to hide the patient's information.

5.1 MOTIVATION

As mentioned, the adoption of EMR systems has been much slower than expected. This becomes immediately evident when one compares a conventional paper-based medical record (Fig. 5.2(a)) with a typical commercial display of an electronic medical record (Fig. 5.2(b))—the information organization is rather similar! There are still separate boxes with textual information, they are just now in form of tabbed windows which can be scrolled and clicked on to obtain more detail. Indeed, handwritten notes are now replaced by easier-to-read printed text and browsing through paper document folders is replaced by more convenient scrolling and mouse-selection operations. This clearly is an advance, also in terms of portability, but the opportunity to reformat the digital information into more effective displays is largely being missed, and this leads at least partially to the current frustration with EMRs. Furthermore, a significant problem is also that there are no provisions for scalability—an increase in data and information simply leads to more scrolling and more diverse and deeper selection hierarchies.

The existing problems with current EMR systems have been rigorously studied in [163] from a usability standpoint. The study finds that the key principles such a system should obey are simplicity, naturalness, consistency, minimization of cognitive load, efficient interactions, forgiveness and feedback, effective use of language, effective information presentation, and preservation of context.

We believe effective and robust information organization and integration via well-established criteria is a key to achieve these requirements. A hierarchy is a convenient data structure for this purpose, and the standard codes commonly used for billing in hospitals offer such a robust and hierarchical information organization. These codes are ICD (International Classification of Diseases), CPT (Current Procedural Terminology), and NDC (National Drug Code). ICD describes the condition or disease being treated—the diagnosis. CPT describes medical services and procedures performed by doctors for a particular diagnosis. NDC codes the administered drugs. As an added benefit, by building our visualization framework on top of this ubiquitous medical code infrastructure we also facilitate a seamless integration into existing hospital systems which use these codes ubiquitously to index medical facets.

5.2 THE FIVE W'S SCHEME

We shall first discuss the conceptual information organization of our system, in terms of structuring the Five W's.

5.2.1 THE WHO AND WHAT

The *who* and *what* information helps doctors to quickly assess the history and status of the patient. It describes the patient in terms of:

◆ Symptoms and Diagnosis: include the patient's symptoms, injuries, and any diagnosed diseases. All of this information can be encoded using the ICD code.

♦ Procedures and treatments: include patient tests and examinations, treatments administered, and drugs prescribed. This type of information can be encoded using the CPT code or the ICD-procedure code and NDC code.

♦ Data: include test and examination results, review of systems, vital signs, and social and family history. The codes for these are part of the procedure code and yield information on what the patient already has.

♦ Temporal information: a time stamp or interval.

♦ Severity: a value characterizing deviation from normal.

Our system encodes this information in two ways— hierarchically organized by medical codes and sequentially in form of relations and causations.

5.2.2 THE WHERE

The *where* information refers to the location of the *who* and *what* information within the confines of the patient's body. While not all information can be localized that way, for the information that can, we encode it in a body outline map onto which information items are linked to their appropriate body locations.

5.2.3 THE WHEN AND WHY

The *when* and *why* show a case under (doctor) collaborative diagnosis/treatment, or an entire life span. This is best conveyed by a sequential chain that emphasizes causal or temporal ordering. Such a chain stresses causal relationships and encourages causal reasoning done by the physician. It also aims to model the standard medical workflow: (1) observe symptoms and possibly browse history data, (2) prescribe and evaluate tests results, (3) form hypotheses and possibly acquire more data, (4) cast diagnoses and (5) prescribe treatments. These steps may all be executed within one patient visit or they may prolong over some period of time, but the overall workflow is always engaged. The 5th step may include a referral to another doctor, which then starts another workflow (back-linking to the previous).

The *why* is represented by relationships. Doctors have the option to create links between different medical entities, using their medical knowledge. For single chains the system simply connects the event chain one by one.

5.3 ENCODING THE FIVE W'S

Fig. 5.1 shows our system's user interface along with the two types of cooperating displays it offers. In the following, we first provide an overview and then discuss each display in detail. The displays are:

♦ A hierarchical radial (patient overview) display (Fig. 5.1(a)) with an integrated body outline primarily for the *who* and *where*. It allows doctors to quickly survey and focus on details

of the patient's medical history in a fact-centered and anatomy-referenced fashion, presenting symptoms, diagnoses, procedures, treatments, and data along with a *time occurrence histogram* (the *when*).

♦ A sequential (diagnostic reasoning) display (Fig. 5.1(b)) primarily for the *when* and *why*. It enables doctors to see and augment the medical records in the context of the diagnostic workflow—visit, symptom, test/data, diagnosis, and treatment.

The *what* is part of both displays (in form of the various nodes) and is context-sensitive. The two displays are linked, such that operations on either one will be reflected in the other. Thus, one can quickly switch between the (possibly evolving) sequential diagnostic reasoning flow and radial overview displays. The radial display is also able to communicate causal relationships, but in the context of the entire history of the patient. Our user interface provides various facilities for filtering, sorting, selection, and searching, which are available for both displays.

5.3.1 HIERARCHICAL RADIAL DISPLAY

There are in fact three radial displays, one for symptoms and diagnoses, one for procedures and treatments, and one for data Each uses the appropriate standard medical billing codes as an organizational model. For example, the 'symptom and diagnosis' display is organized by ICD9 code, a very detailed and readily available medical hierarchy. We are currently adapting our framework to the new ICD10 code which is an expansion of the ICD9 code.

We use a tree data structure to store the code hierarchy information. For each symptom or diagnosis the patient has, we find the node *n* in the tree with the corresponding ICD9 code, and insert the new item as a child for node *n*. For example, if the patient has *bacterial meningitis* (ICD9 = 320), we first build an incident (medical facet) node *m* for this diagnosis to store its information (time, severity, result, etc.). In the tree, we find the node *n* with code 320, which is [320 *bacterial meningitis*]. Then we insert *m* as a child of *n*. Next we update all ancestors of *n* with the new inserted incident node's information, such as number of incidents that fall into this category, severity, etc. This also updates the time history histograms. By doing this for all symptoms, diagnoses, and procedures, the tree will always be current and contain the patient's entire history.

Visual Design

There are many methods to visualize hierarchies [34]. We chose a space-filling paradigm because it can be better restricted to occupy a given space than overlapping visualizations, such as node-link diagrams. For space-filling visualizations we had the choice between rectangular and radial displays. Treemaps [65] is a popular member of the former category. We ultimately chose a radial one—the sunburst [125]—because it allowed us to easily integrate a body map into the center and so make the map equally accessible to all nodes. This presented a clear justification for us to use a radial over a Cartesian layout which has been shown by in [33] to bear some advantages in terms of accuracy and ease of reading. On the other hand, we do follow the study's other guideline—to encode the more important dimension in sectors (as opposed to

rings).

The sunburst is a radial hierarchical space-filling diagram. Nodes in the sunburst layout are drawn as solid areas (either wedges or bars), and their placement relative to adjacent nodes reveals the relationships in the hierarchy. Because the nodes are space-filling the angle for each node can be used to encode additional information, such as number of incidents in our case. The sunburst layout has been widely used to visually encode hierarchical structures, such as documents [29] or software systems [78].

Show Where

Typically, the root of the tree is displayed in the sunburst center. However given the sole application context—the patient—we replace the standard root node by a human body template. This enables us to intuitively fuse the *who* with the *where* display. The two displays are interlinked, such that nodes in the sunburst point to the appropriate body locations (if such a mapping exists). If an incident (medical facet) has corresponding location information, a red dot is displayed in the body outline; otherwise, it is mapped to a dot outside the body outline (above the head). The intensity is used to encode the severity, which is computed using the same color composition method than for the nodes (see section on Color Design below). Thus by looking at the body outline, doctors can quickly learn which parts of the patient's body have (or had) diseases and also judge their severity by the color intensity. Hovering on the red dots will popup more details about the injured part, such as name, severity, and how many incidents are related. Clicking the red dot will highlight the corresponding diseases in the sunburst tree.

Node Design and Time Histogram

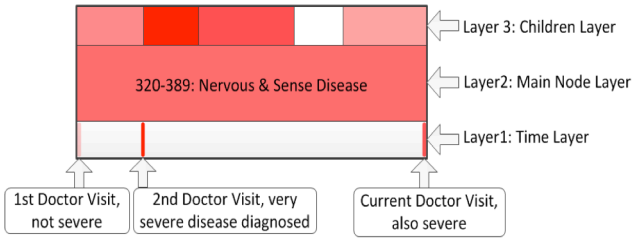


Fig. 5.3: Node design. Shade of red encodes severity. This node tells us that the patient has a relatively severe disease in the nervous and sensory organs. There have been a total of three doctor visits related to these diseases. The children layer gives more detail on the specific kind of disease within this broad category.

Each sunburst node has a wedged shape. We further decompose each node into three layers to encode more information, as is shown in Fig. 5.3. These layers are:

♦ *Layer 1 (Inner layer):* encodes time information. The length of the node represents the entire patient history, that is, going from left to right (clockwise in the radial layout) we encode the history from the first doctor visit up to now. This *time histogram* allows doctors to see how often the patient received a certain treatment or exhibited a certain symptom, or how long ago this occurred, etc.

♦ *Layer 2 (Middle layer):* the main node layer which is used to display information about the node, such as code, name, etc.

♦ *Layer 3 (Outer layer):* encodes the next lower level in the hierarchy. It is meant to give users a quick overview on the sub-diseases without showing their real nodes. We provide this

layer to make the hierarchy display scalable.

Integrated Display

The three hierarchical radial displays, symptoms/diagnosis and procedures/data, and treatments can be combined to display the entire (medical) picture of the patient as well as the relationship between them. The corresponding sub-rings are colored accordingly. If two incidents are related with one another, an edge is drawn between them. With these relationships, users can select a node to see what other nodes are related to this node. This exploratory functionality can assist doctors in the medical reasoning process. In order to get all dependent nodes for one selected node, we use a graph traversal algorithm to compute the dependent closure.

A First Example

Fig. 5.1(a) combines the diagnosis view (left half with dark red ring) with the test/data view (right half with green ring)—the color legend is in the visualization controls panel on the right. In the example shown here, a patient visits the doctor complaining about blurred vision. This is the first visit of many and we will return to this case in Section 5.5.2. The body map in the center shows the anatomical location (here the eye) of the patient's medical problems, and the edges point to the corresponding nodes in the radial display (here from test "eye exam" to diagnosis "retinal inflammation"). Doctors can click on any node to obtain more detail and can then pin this information to the display as "post-it notes" (see Section 5.4.1). The time histogram in the inner node layer has only one bar marker on the far end since this is the first visit of the patient—else there would be more bar markers distributed over the layer ring.

Color Design

When refining our display with our collaborating doctors, we were repeatedly told that one feature they cared very much about was the ability to quickly assess the severity of a symptom or disease. Therefore, in all three layers, the shades of red encode severity information—full red encodes highest severity 10/10 and white encodes no severity 0/10. We used the 0-10 scale because it is often used in the social sciences and in medicine, for example the Comparative Pain Scale [164].

We use the linear red color scale to shade severities in between 0 and 10. We may also encode severity on a diverging scale—severely low and severely high—color-coded using an appropriate diverging color scale [19]. We employ green-white-red to signal positive and negative outcomes, respectively.

If the node contains multiple incidents in its children, we use color composition to summarize this information. Our interface provides two color composition techniques:

- MAX: takes the maximum value from all composited severities as the current node's severity and uses it to compute the color. This composition means that if there is one sub-disease/sub-symptom that has high severity, then its parent category should also be paid attention to.

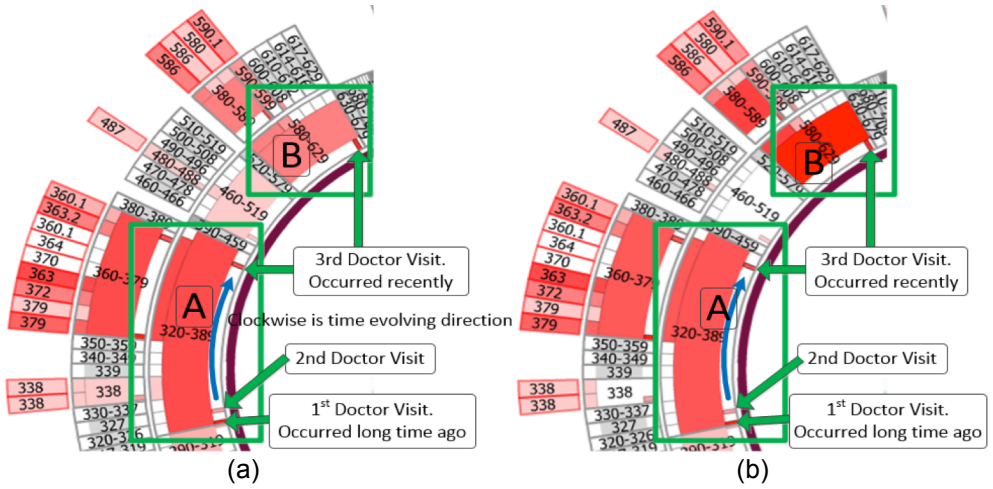


Fig. 5.4: Color Composition. Node A contains three doctor visits; Node B contains only one but very recent doctor visit. (a) Using the MAX operator. (b) Using color composition. The coloring of (a) suggests that node A has the highest priority. But A's color is determined by a severe disease which however occurred in the patient's first doctor visit (a long time ago). That disease might not be as important compared to some less severe diseases that occurred in the patient's last visit (just a few days ago). So, if more recent occurrences are the focus, this disease will be better highlighted using color composition (shown in (b)) which takes time into account. This makes node B more apparent.

♦ Compositing: computes the color using composited rendering along time. This method fades the colors for past events. Since only one color (red) is used, the color composition can be solved by an alpha blending equation:

$$S = max\{\sum_i (S_i \cdot w_i);\ 1\} \tag{5.1}$$

Here for each incident i, S_i is severity and w_i is the corresponding weight, which is a function of time. Early incidents have lower weights and more recent incidents have higher weights.

Fig. 5.4 shows via an example that each technique has its own advantages and disadvantages. The MAX operator can draw a doctor's attention to the most severe diseases, no matter if they occurred a long time ago or just now. This can be good for some long-term severe diseases, such as diabetes. But it may cause misunderstandings for gradually recovering diseases, such as bone fractures. Conversely, the color composition technique takes time into account, it fades diseases that occurred a long time ago and highlights the most recent ones. The two modes are complementary to each other—we observed that our collaborating doctors switched back and forth between the two modes when exploring a medical history.

Edge Coupling

Edges in the integrated view are displayed according to which level of the hierarchy is chosen. Consider Fig. 5.5 where we show a pair of related nodes in an integrated display and three code levels. The original edges *e1* and *e2* link the nodes corresponding to the specific

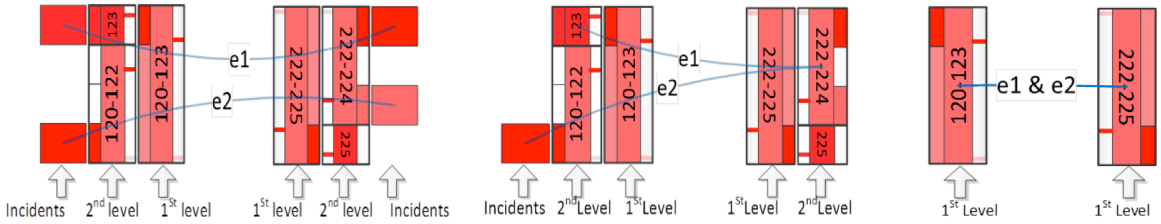


Fig. 5.5: Node collapse strategy. (a) Original edges e1 and e2 link the incident nodes; (b) Node [222-224] and [123] collapse. Edges linked to their children now link to themselves. (c) Only the first level nodes are shown, then e1 and e2 merge into one single edge. The opacity of the edge is computed by compositing those of e1 and e2.

incidents (the leaf nodes) of level 3 (Fig. 5.5(a)). As the user collapses these incident nodes, the edge will link to their parent node (Fig. 5.5(b)). If we collapse all of the incident nodes, then *e1* and *e2* will be merged together (Fig. 5.5(c)), and the corresponding intensity of the edge increases. Edge bundling [57] is used to reduce cluttering.

Interaction Design and Scalability

Each radial display is either hierarchy-centric or patient-centric. In the hierarchy-centric display (Fig. 5.6(a)) each node in the sunburst tree is sized by how many sub-categories it has. It focuses more on the hierarchy information represented in the medical codes and serves as an illustration of the complexity of a sub-system and its composition. Conversely, in the patient-centric display, more radial space is dedicated for diagnoses/procedures the patient had activities in. For categories that the patient does not have any activities in, the node will be collapsed to

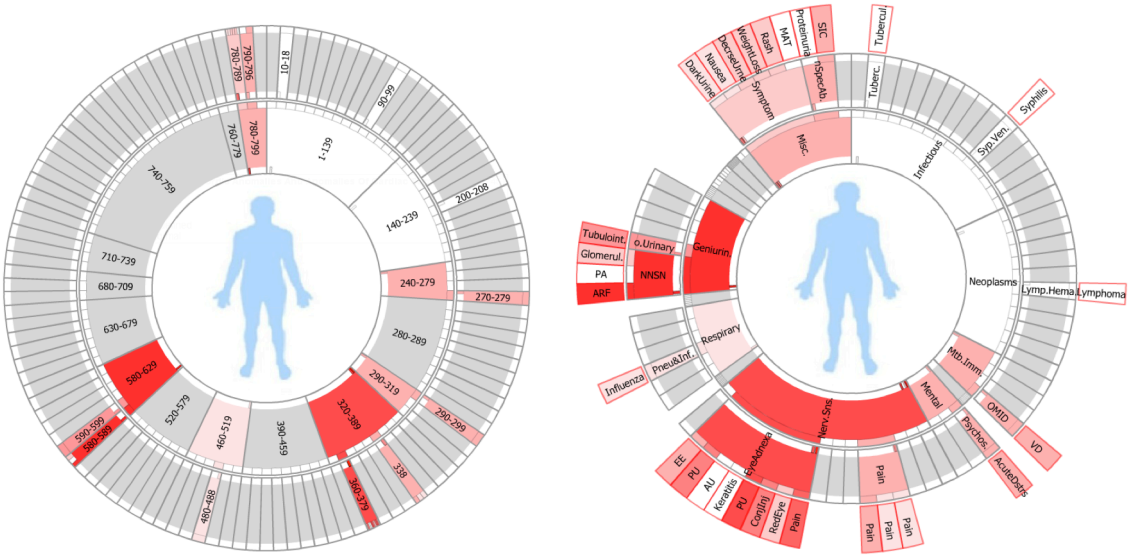


Fig. 5.6: Different layouts for the sunburst display with two alternative node labeling schemes. (a) The hierarchy-centric layout. The nodes are labeled with the ICD9 codes but labeling with the corresponding medical terms is also possible. (b) The patient-centric layout. Here the nodes are labeled with the corresponding (abbreviated) medical terms as available in the ICD9 description.

save space for others (see Fig. 5.6(b)). This display is most often used as it makes better use of the space.

Multi-Level Interaction

The sunburst radial display in Fig. 5.6(b) shows three levels of the code hierarchy. Level 1 corresponds to the highest code hierarchy level. Level 2 shows more detailed categories. Level 3 contains the incident nodes, which are the actual medical facets (symptoms/ procedures/diagnosis) that the patient has activities in. The user is given the choice to either display the medical code or the corresponding term in each node (see Section 5.3.1-Node Labeling). The first level always shows codes/names to provide an overview. Likewise the leaf nodes also always show the codes/names for detail. The middle levels (2^{nd} level) show codes/names only when they have incidents (children).

Three default level filters are provided to help users quickly explore these three levels. Users can expand and collapse the nodes interactively. In ICD9 certain conditions can have up to 5 levels and ICD10 has even more. Our sunburst display is scalable to support these additional levels by extending the underlying data model.

Augmenting the Display with Post-It Notes

Hovering over any node will reveal more information on a yellow *post-it note*. Clicking on the post-it note will pin it to the display for fast recall (Fig. 5.1(a)). Using them within a *diagnostic sandbox,* doctors can focus on the post-it noted symptoms/exams/diagnosis. The post-it notes can include data as well, such as an image or a time-plot. We have two versions for each: a thumbnail for pinning and the original size for exploring.

Filtering, Transformations, and Zooming

A time and severity filter is provided to filter out unrelated or unimportant incidents. For example, doctors can specify a time range and a severity threshold. Then only the incidents that occurred during the specified time range with severity values higher than the specified threshold are shown.

Further supported user interactions help users explore the patient information and see details on demand:

- Translation, zoom, and rotation: these interactions manipulate the radial view to put the important features in the center of the window. We found that this helps users to stay focused and promotes ease of reading.

- In the integrated view, users can zoom into one specific hierarchy display by making its angular range larger. This shrinks the angular ranges of the other categories and all nodes are resized accordingly. Like the expand/collapse feature, this interaction helps users to focus on a specific disease of interest.

One concern about these interactions is that they might destroy a doctor's mental map. However, our user study (see Section 5.6) indicates that most doctors are comfortable with this feature. Also, users can always choose not to use these features if they dislike them.

Node Labeling: Medical Code and Terms

The display nodes can be alternatively labeled by the ICD code number or by the corresponding text (as shown in Fig. 5.6). The interface provides a button by which a user can quickly switch from one representation to the other.

A challenge is to find good abbreviations for the nodes' labels. The ICD9 medical terms can have very long strings. For example, code "049.9" stands for "Unspecified non-arthropod-borne viral diseases of central nervous system". To make the display visually manageable, we aim for a string length of 10 characters or less. This ensures that no label extends much beyond its node's border. We found that while some standard medical abbreviations exist, for example, *CNS* for "central nervous system", abbreviations for others are less obvious. Our current approach uses standard techniques to shorten the strings. First we remove all stop words such as "the" and "of". Then, depending on the length of the remaining words we employ the following increasingly aggressive techniques:

- Contract: omit some or all interior portions of the word but retain its first and possibly last letters or elements.
- Abbreviate: cut the word after some characters and then terminate the remainder by a period.
- Acronym: only retain the first character of each word, turn them upper-case and concatenate without blanks.

These strategies have helped to shorten all strings to the desired length of 10 or less. Yet, the result is not always satisfactory. We have therefore devised an interface that doctors can use ad-hoc to define a better abbreviation for a term that they think is not well represented.

5.3.2 Sequential (Diagnostic Reasoning) Display

The sequential display is used mainly to demonstrate *what*, *when*, and *why* information. Usually the diagnostic workflow is: Patient visits doctor → patient complains of symptoms → doctor orders tests → doctor renders one or more diagnoses (valid or not) → treatments are given → outcome is observed. Thus, a sequential display can show this reasoning chain very well.

The medical records are organized by an underlying graph data structure. Each node corresponds to one stage of the workflow—visit, symptom, test/data, diagnosis, and treatment. Edges represent relationships. Each stage corresponds to a column of equal-colored nodes. This achieves a well-structured design that reduces the overhead for visual search and therefore lowers cognitive load. An earlier version of this display [151] used a force-directed layout in which the various types of nodes could appear anywhere on the canvas and were only distinguished by color. We abandoned this display since our medical collaborators found it difficult to work with.

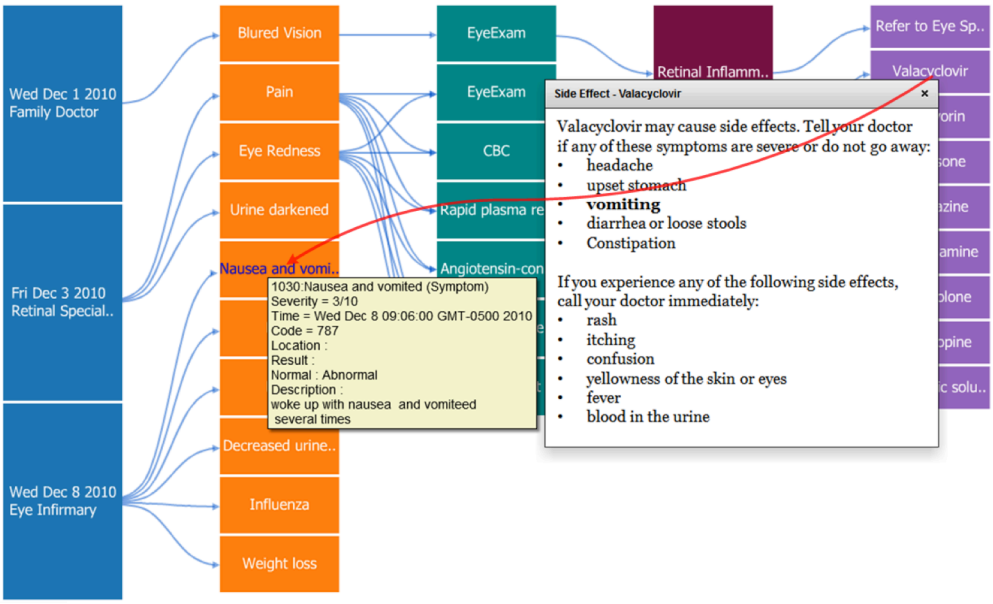


Fig. 5.7: Some features of the sequential display: information text window, post-it note, and red back-edge. A back-edge from a treatment to an incident can denote a referral, while a back-edge from a treatment to a symptom can denote a side-effect as is the case here – the drug *Valacyclovir* prescribed at a previous visit (Dec. 3) causes the current visit's (Dec. 8) symptoms of nausea and vomiting.

Visual Design

A node is displayed as one elongated box—we found that this better utilizes the rectangular screen, better fits the text, and has better scalability compared to a circular shape. All of our medical collaborators agreed on this. If two nodes are related with one another, an edge is drawn to link them together. As an example, a very simple sequential display is shown in Fig. 5.1(b). It incorporates the test and diagnosis nodes of the radial display of Fig. 5.1(a), and the other elements of the visit.

More complex chains can have many edges. We use edge bundling to reduce the cluttering that may occur. Further, in some cases the current doctor refers the patient to see another specialist (which is the treatment in this case), or current symptoms are caused by previous described drugs (which can be a form of diagnosis). In these situations back edges appear. Back edges are shown in different color (red) to make them easy to see. Fig. 5.7 shows an example for a back edge drawn to indicate a side-effect (see caption). We find that the equivalent radial display shows the back edges especially well because they go against the flow of (cross) the other edges. Our system currently does not have a specific column or radial display for treatment outcome, as a gauge of effectiveness. Rather, outcome is logged and can be monitored by examining the corresponding treatment node which is back-linked to the other nodes in the diagnostic workflow.

Fig. 5.8: Severity and uncertainty rating bar variations for sequential display nodes. The location of the vertical black line below the node indicates whether the data item is a two-sided rating (black line in the middle) or a one-sided rating (black line on the left). The length of the bar encodes the value and its saturation encodes the uncertainty of that value. The text inside the boxes above explains the semantics that our system uses in more detail.

Rating

All incidents can be rated by the physician on the fly using a popup with a slider. To encode the rating, one option is to use the same method as was used in the radial display—color. This would make for a consistent encoding. However, there is a conceptual difference in how the two displays are used. The radial display is meant to provide an overview where color (in particular shades of red) can quickly guide the viewer to the more severe nodes. Conversely, the sequential display is for diagnostic reasoning where quantitative assessments are to be made. According to Bertin's levels of organization [14], color and brightness are selective and ordered but only size is quantitative. With this in mind, we use different types of visual severity encodings for the two displays: color (saturation) in the radial display and length (of a rating bar) in the sequential one. This rating bar is positioned below the corresponding display primitive and the semantics of the ratings determines its color and variation. Fig. 5.8 illustrates the various schemes which we will further explain below. The rating uses standardized levels to gain independence from scaling issues and so provide for a scale-neutral node display. Our medical experts indicated that they are able to translate these ratings into actual values using their medical knowledge. But, hovering over the node will display the actual values.

Symptoms, Tests, and Diagnoses are rated in terms of severity, that is, the deviation from normal according to some scale. As mentioned we adopt the Comparative Pain Scale [164] of 0-10. Severity is encoded in a *severity bar*, which is grey with full length at first, meaning the node has not been processed by the doctor. After the doctor looks at the node and sets its severity or normality, the severity bar will have the same color as the node, which our doctors agreed to be the most aesthetic. The bar's length is weighted by the severity level. Our system supports two types of severity:

◆ Two-sided (often used for data): the normal value is in the center and deviations are either too low or too high.

◆ One-sided (rates severity of symptoms and diagnoses): the normal value is on the left and the most severe value is encoded as a bar with full length.

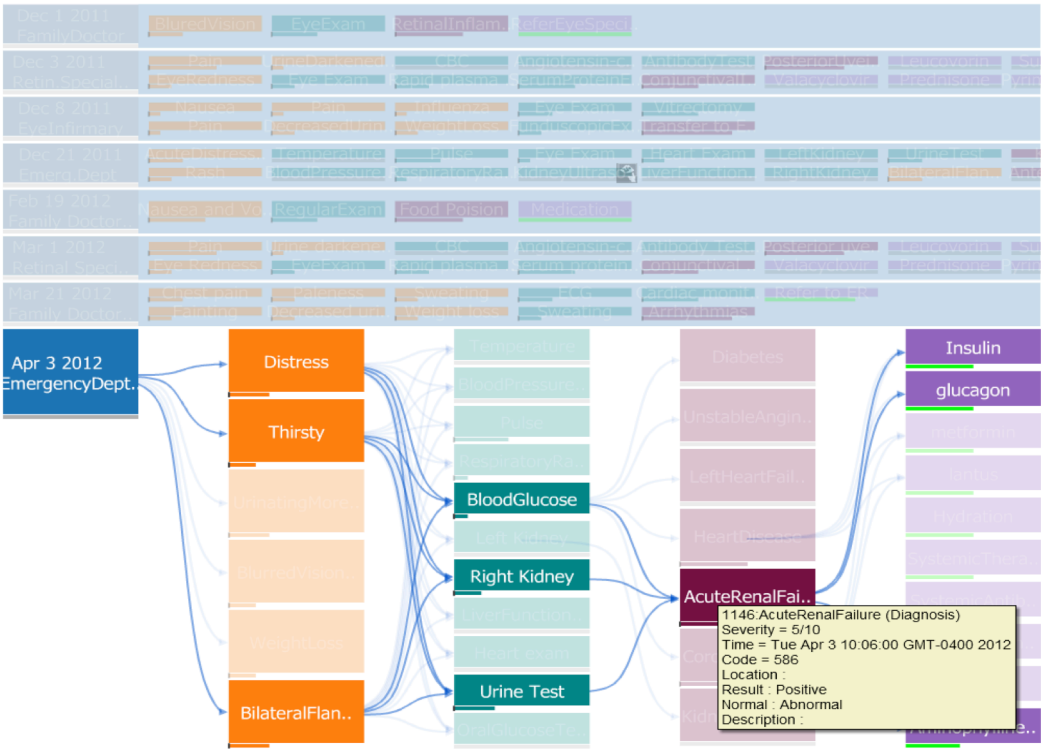


Fig. 5.9: Sequential (diagnostic reasoning) display. (i) Node-collapse to focus on the most recent visits – a total of more than 100 incidences is shown. (ii) In the browsing highlight mode, the doctor has selected one of the diagnoses which highlights all related nodes and branches but fades out the others. The same type of highlighting also occurs when filtering.

Treatments are rated by their outcome—whether the treatment has a positive (successful) effect on the patient or a negative effect (unsuccessful, causing side effects). Green color encodes positive effects, and red encodes negative effects. The length of the rating bar means how successful/poor the positive/negative effect is.

Uncertainty

The uncertainty or confidence in a diagnosis or test is encoded as the saturation of the corresponding node—full saturation means no uncertainty (full confidence) and low saturation means high uncertainty (no confidence). For example, if the doctor thinks that the patient may have a 50% possibility of suffering from lung cancer, then the saturation of the node is reduced by 50%. This rating is also given by the doctors interactively.

Interaction Design and Scalability

Hovering over a node brings up scrollable text windows that provide relevant information, such as data details, side-effects of a treatment, narrative text from a patient report, and so on. Also, similar to the radial display, “post-it notes” with a more detailed description can be pinned at any location, possibly reduced to just show a few pieces of information, such as the rating and the full name of a disease (which appears as an abbreviated label in the node). Finally, any data associated with a node can be brought up by clicking on the corresponding data icon in the box.

The nodes may be sorted by any data field: time (date), doctor, severity, etc. Scalability is achieved by (1) muting unselected nodes and their links and possibly completely collapsing them, (2) aggregating related nodes into a single box where these groupings can be defined by similarity in the sorting variable, such as data, temporal, doctor, and the like (one typical grouping might be a visit with a specific doctor), and (3) filtering with a global slider by severity, time interval, etc. Fig. 5.9 presents an example.

We use edge bundling to reduce clutter. Since back edges are drawn on top of other nodes/edges, too many of these tend to clutter the display. Hence, the back-edges are not shown by default. Users can turn on those edges to see the transitions. Also, in browsing highlight mode (Fig. 5.9 (ii)), users can hover on any node and then only the nodes that are related to the selected one and their corresponding edges will be highlighted. The same type of highlighting also occurs when filtering is applied. The effect is real-time and it allows users to quickly browse complicated displays by moving over nodes and branches which then highlights the associated elements and muting others. Our video demonstrates this function in action.

5.3.3 Implementation Details

The user interface is implemented using Action Script and the Flare visualization toolkit [162]. The backend server is implemented in Java and Java EE. It connects the front-end interface with the structured database that stores the patient histories. Each patient record contains: visit ID, type, code, description, time, etc., and if available, severity, uncertainty, body location, and IDs for related reports or diagrams. The relationships are built by the physicians during the exam and are stored in a separate table. The communication between the interface and the server is achieved with the help of BlazeDS.

5.4 Integration into Hospital Workflow

We see at least two places in a hospital workflow in which our system can proof useful: (1) as a diagnostic assistant in the patient-doctor encounter to help doctors learn about the patient history and support clinical decision making, and (2) providing medical coding support in the hospital's coding and billing departments.

5.4.1 Diagnostic Assistant

When doctors perform a diagnosis, it is essential to have a good understanding of the patient's history. Further, one doctor often takes charge of several patients, particularly in busy emergency room scenarios. Therefore, the smaller time that is spent on learning a patient's history, the more efficient actions will be taken. And as a result, the more patients will be taken care of. Our system allows doctors to quickly get insight into important issues like:

- What were the most severe symptoms this patient had, now, recently, and in the past?
- What tests have been done related to these symptoms, and what were the results? Were there treatments that the patient did not respond well to, or not at all?

♦ What were the diagnoses rendered in these tests? What were the outcomes? What were the reasoning chains that led to these diagnoses? Were there ruled-out diagnoses?

♦ What medications were prescibed in the past and when? What side-effects do they possibly have, and might they have something to do with the present symptoms?

With respect to the last issue, since our system is connected to online databases, a doctor can quickly research information on drugs and their side effects and other information on possible treatments and causes.

5.4.2 MEDICAL CODING SUPPORT

Medical coding is the transformation of report-based narrative descriptions of diseases, injuries, and medical procedures into numeric or alphanumeric designations—the ICD, CPT, and NCD codes—that are used to bill patients and insurance. Hospitals typically have certified staff for this task: medical coders. Medical coding is not without challenges, and we shall list the most relevant next:

♦ Poor doctor's handwriting: this challenge is trivially overcome by computer-based input.

♦ Mismatched terminology: cases can exist when doctors use a different terminology than the one formulated in the ICD or CPT codes. So coders looking at an operative note might expect a certain descriptive word for a procedure and if they won't find it, they will code that the procedure was not done. However, the doctor might have described the procedure in different terminology, and so the procedure would go unbilled.

♦ Unbundling: this describes the fraudulent process of breaking apart (fragmenting) codes that are inclusive of other codes. An example is coding two units of CPT 67311 (strabismus surgery, recession or resection procedure, 1 horizontal muscle) instead of one unit of CPT 67312 (strabismus surgery, recession or resection procedure, 2 horizontal muscles).

♦ Upcoding/undercoding: the former is the fraudulent practice, in which provider services charge for higher CPT procedure codes than those actually performed, resulting in a higher payment. Since the rules are fairly complex, just to be safe a doctor may deliberately bill for work on lower level codes even if more services were performed, leading to loss in revenue.

♦ Not coding the diagnosis to the highest level: some ICD codes need a 4^{th} or 5^{th} digit to be accurate and correct, many coders tend to use the highest level to save time.

♦ Incomplete reporting: physicians may not report on everything they did although they may have performed it.

Since users can easily switch between medical terms and ICD codes for the nodes, our system can be used by both doctors and coders—the underlying (medical code-based) hierarchy is identical. Doctors tend to be less familiar with the actual ICD codes and so they typically use the medical term labeling almost exclusively. Coders on the other hand make use of both medical terms and medical codes. By using the same display infrastructure for both reporting and billing

the possibility of the problems due to mismatched terminology is greatly reduced. Further, our system also provides medical coders with a much better overview about the services performed and what services may have been performed and so helps avoid revenue loss due to incomplete reporting. Coders can quickly and better see relationships of treatments and procedures and so avoid upcoding, undercoding, unbundling, and other reporting errors that often lead to lengthy and costly struggles with insurers.

5.5 USAGE SCENARIOS

We have explored a few usage scenarios to demonstrate the effectiveness and efficiency of our system. One scenario is reconstructed from a complex medical case involving a number of physicians. The others are based on routine medical cases that occurred at our home institution.

5.5.1 SCENES FROM DAILY CLINICAL PRACTICE

For this part of the study we identified four sample scenarios from daily emergency room (ER) practice. They were compiled by our collaborating ER physician who has 25 years of professional experience. This ER doctor has long been looking for an interface where "information is right there when I need it" and came to us highly frustrated with the current state of the art.

We compare our prototype with a state-of-the-art commercial EMR system. Here we are mainly interested in gauging how efficient each can provide insight into a patient's medical situation. As a quantitative measure for this capability we count the number of mouse clicks needed to uncover a specific piece of information [163]. We analyze the four scenarios using both our prototype and the commercial system from the hospital (with similar screens in Fig. 5.1(b)), accessing a (patient de-identified) copy of our university hospital's database. For the latter two scenarios we only briefly summarize the possible interaction without figure references. In the following, we shall motivate each scenario by a specific clinical task.

Diagnostic Medical Reasoning

We choose a patient who has been admitted to the ER with serious nausea and irregular heartbeat. A test result points to a low potassium level. A deeper look at the patient's recent history reveals that he was diagnosed of congestive heart failure (CHF) in the past, and prescribed *Lasix* as a preventive medication. The doctor knows that low potassium level can be a possible side effect of *Lasix*, and so the current diagnosis could be related to it. An alternative medication is prescribed. To obtain this information from the hospital's health IT system the required number of mouse clicks n_c is (at least) 9**:**

n_c = [go to patient details] + [problems & diagnoses] + [medication list] + [first med.] + [details first med.] + [second med.] + [details second med.] + [third med.] + [details third med.] + click X different rows = 9 + X

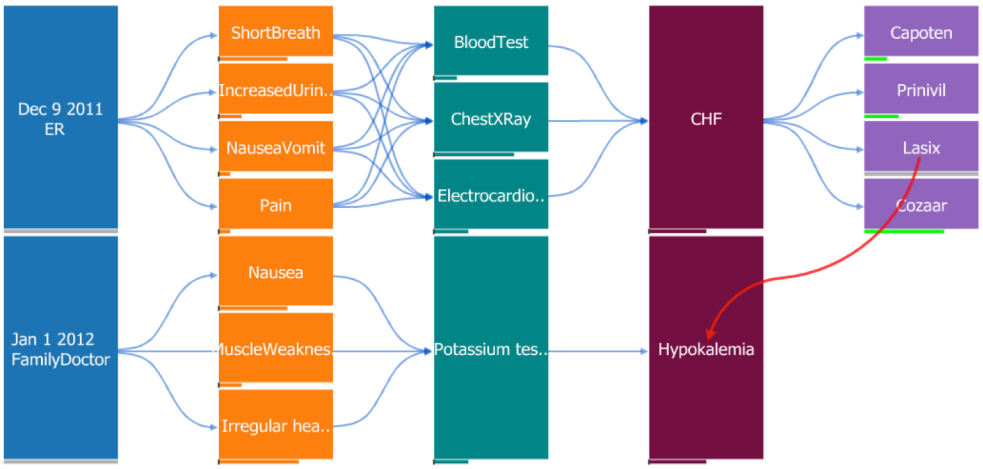


Fig. 5.10: Diagnosing a case of *Hypokalemia* with the sequential display. The cause is a recently prescribed medication: *Lasix*.

Each mouse-click changes the screen, breaking the mental flow. Conversely, in our system the doctor has the choice of using either the radial or the sequential display. Fig. 5.10 shows the latter, with just the most recent patient visits summarized. With one glance the doctor learns about the patient's CHF history and the four medications that were prescribed. Seeing *Lasix* as one of the medications on the list and connecting this fact with the current finding of hypokalemia (low potassium), the doctor quickly gets the answer. A red back-edge is drawn to indicate this causal relationship, and an alternative medication is prescribed (not shown). Hence, with our system the number of mouse clicks is just 1 (to select/filter the recent events from the sequential display). No screen ever changes—the doctor maintains full overview at all times.

Detection of Substance Abuse

Back-pain is frequently reported in the ER, and narcotics are often prescribed without much examination of patient history since it takes too much time with current EMR systems. However, at many occasions, patients either simulate their back-pain to obtain narcotics for street sale or own personal abuse, or they have fallen victim to chronic pain which should be treated via alternative ways.

Fig. 5.11 shows the radial displays for two patients, A and B, who both complain of severe back pain and request narcotics to relieve this pain. Back pain falls into the large category of *musculoskeletal and connective* (labeled *Mus. Conn.*). By looking at the time histograms the doctor quickly sees that patient A did not have any back pain before, while patient B has had regular hospital visits for chronic back pain. The appropriate courses of action are now taken. Obtaining this insight with our interface took a simple glance at the radial display. For the commercial system, on the other hand, the same information required $n_c = 6$ mouse clicks.

Assembling and Connecting Information

Another problem with current systems is that related information is difficult to connect and assemble. For example, patients frequently carry dozens of medications, prescribed by different

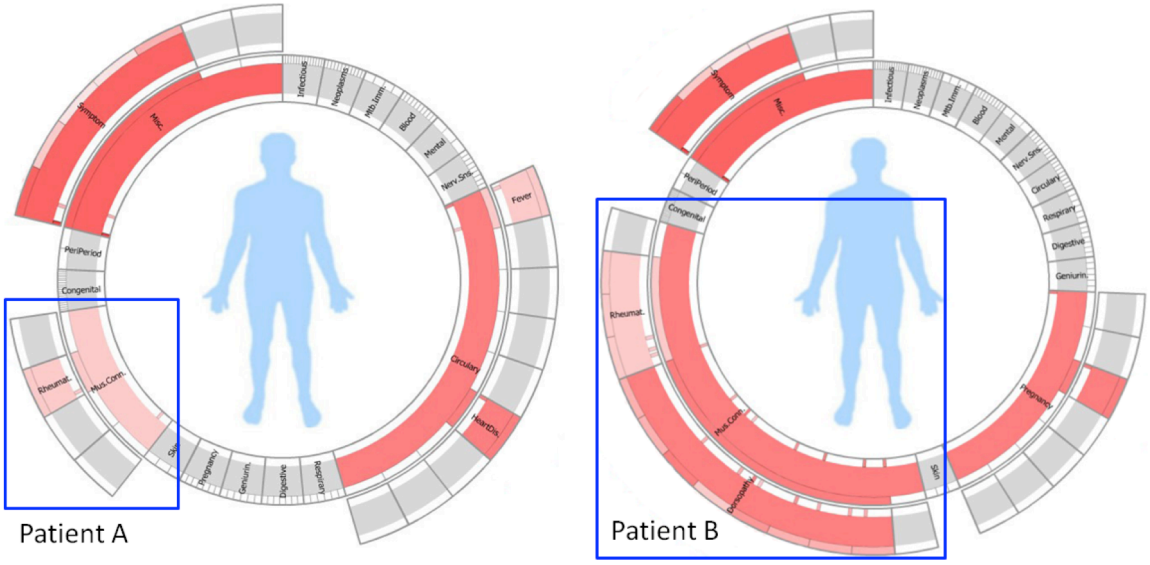


Fig. 5.11: Radial displays for two patients reporting to the ER with back pain. The relevant area is the lower left quadrant labeled Mus. Conn. From the time histograms we see that patient A had no incidence of back pain before, while patient B is a frequent sufferer.

physicians. Yet current systems make it difficult to track what a given medication was actually prescribed for. This can have dangerous consequences when a medication has numerous uses (take, for example, Inderal, which treats hypertension, migraine, hyperthyroidism, and angina). These uncertainties often lead to over-prescriptions of medications and unexplained adverse effects. By combining the treatment section with the diagnosis section in our radial display this type of information can be easily obtained in one glance by following the arcs that link two nodes on opposite sides. Alternatively, one could also use the sequential display for this task as well. On the other hand, for the current systems, using suitable examples from the database, we find that $n_c = 6+X$ mouse clicks are required to extract this insight.

Reconstructing Patient History

The following is a more practical case. The ER saw two patients, A and B, both diagnosed with too low heart rates. Both A and B were on lopressors to treat high blood pressure. A routine intervention would have been to give both a pacemaker, but an extensive click-through session with the EMR system finally revealed that A was recently put on a double dose, while B had received the same dose of medication for 5 years. The action was thus to reduce A's dose and only give B a pacemaker. The conventional system required n_c =11 mouse clicks for this, Conversely, our displays are specifically designed to hold such time histories and therefore can reveal them in one view.

5.5.2 Collaborative Analysis

A significantly more complex scenario is derived from a case recently reported in the New

England Journal of Medicine [138]. From it, we constructed simulated visits for the patient (a 22-year old woman) and the four different doctors involved. We then updated the visual displays by the medical information accordingly. In the following we present these displays and some of the interactions that might have occurred if the system had been available during these visits. Fig. 5.12 presents the sequential display that lists the four visits top to bottom, and the radial displays for visit 1, 2, 4, respectively.

Visit 1: Onset—Primary Care Physician

The story begins with the patient visiting her primary care physician, reporting blurred vision in her right eye. The doctor suspects that a retinal inflammation might be the cause and refers the patient to a retina specialist as the immediate treatment. He then annotates this information in the interface, shown in the top row of the sequential display—for clarity we do not draw the back edge of the referral. The radial display annotated with "visit 1" is shown below. The physician marked the diagnosis of retinal inflammation as fairly serious, using the rating popup. The body map has a moderate circle in the eye region, noting the problems there.

Visit 2: Retina Specialist

The day after next, the patient goes to visit the referred retina specialist. She now has severe throbbing pain behind the right eye and also redness. The eye exam reveals conjunctival injection and posterior uveitis. The urine appears to be darkened. CBC and other lab tests, however, turn out to be normal—the displays are updated as the results arrive a few days later. Based on the eye exam the doctor prescribes a number of medications. The radial display is updated accordingly.

The body map now upgrades the eye marking to a full red circle—highly severe—and it also adds a moderate red circle in the bladder region to indicate the slightly unusual urine color.

Visit 3: Eye and Ear Infirmary

Five days later the situation worsens—the patient feels sick again. She checks into the Eye and Ear Infirmary, complaining of her problems—vomitting and nausea which later resolved but was followed by pain in flank and groin. She also mentions her decreased urine outputs and weight loss. From the visual displays constructed so far the doctor quickly learns that the lab tests were normal. Now, for each symptom the doctor needs to find some explanation or devise further tests. This reasoning activity is well supported by our interface. Fig. 5.7 demonstrates the process by ways of the symptom "*Nausea and Vomit*", using the diagnostic chain interface. By looking at the prior chains he notices that the patient's previous treatments contained *Valacyclovir*. He suspects from prior experience that this medication may have something to do with it. To confirm he conveniently pops up the medication information and sees that it has indeed the side effect "*Vomit*". So this may explain why the patient has nausea and vomited, and the doctor draws a red-colored back edge to link the two. He also sends the patient for a vitrectomy. The displays are updated accordingly and the body map now also shows additional problems the patient is reporting, such as a flu.

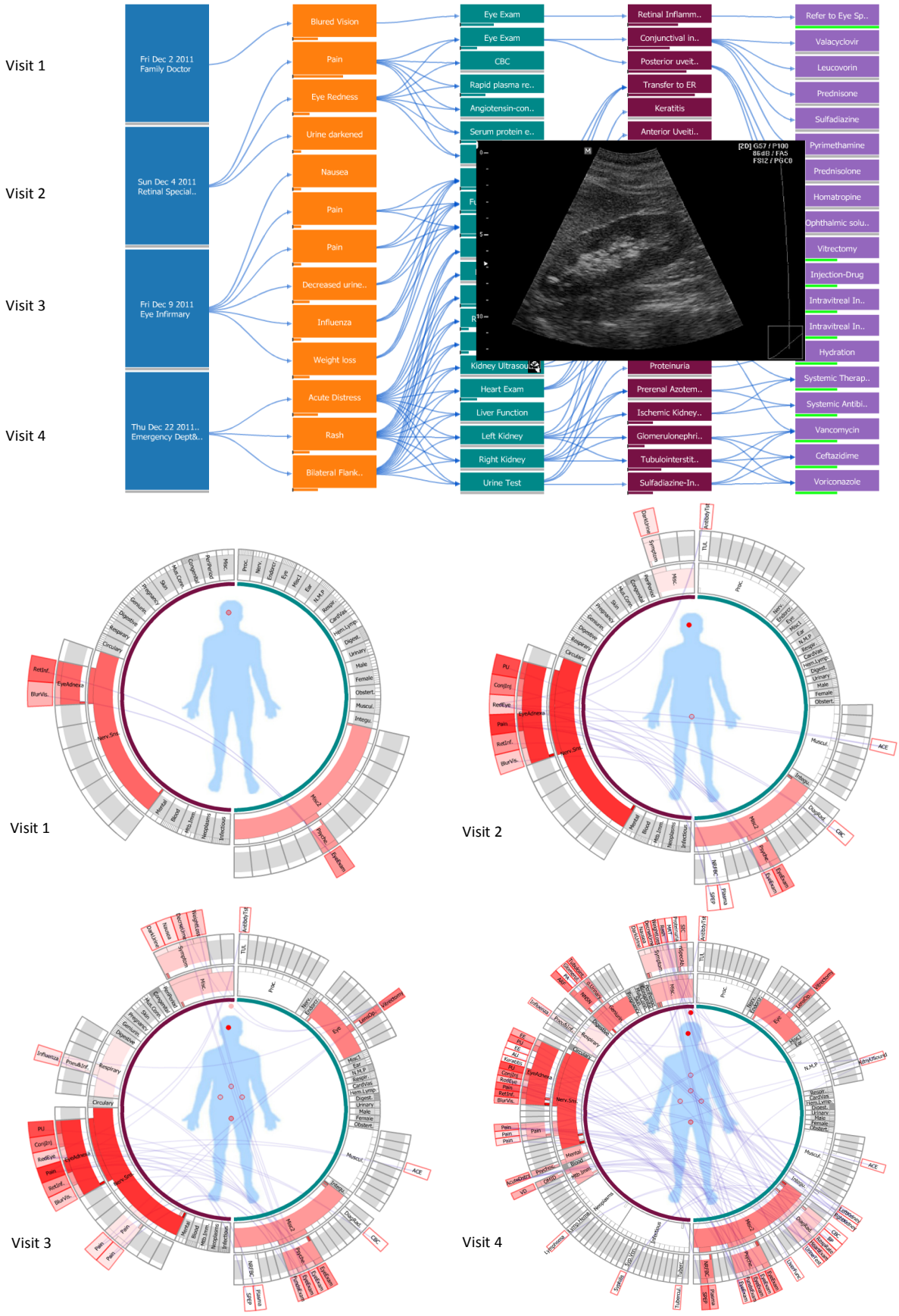


Fig. 5.12: Complex medical case involving four different doctors in a collaborative diagnosis task. Top: sequential display after the fourth visit. Bottom: the emerging radial displays labeled by visit number.

Visit 4: Conclusion—Emergency Department

The case escalates to its peak on Dec. 21, 2011 when the patient reports to the emergency department. Additional diagnoses point to problems with the kidneys—the renal system. The ER doctor assigned to the case takes a renal ultrasound and commits it to the EMR system. Then, as it is often the case in hectic ER environments, he gets called away from the patient to take care of another. A new doctor—a renal system expert—gets assigned and she inspects the displays aggregated so far. She knows that these types of kidney problems can stem from either glomerular, tubulointerstitial, or vascular causes. Looking at the patient history she quickly notices the blurred vision reported recently. This constitutes important evidence in the differential diagnosis that now has to commence—blurred vision is often a symptom in glomerular causes of renal failure. A rapid plasma reagin test is administered but turns out negative, which rules out glomerular causes. Looking back at the sequential display the ER doctor notices that sulfadiazine was prescribed to treat the conjunctival injection—done by the retina specialist on Dec. 3rd (visit 2). This sparks her attention, especially when she sees the flank pain reported on visit 3. She checks if the flank pain could be caused by an obstruction in the kidneys. She uses the sequential display to call up the ultrasound taken by her colleague before (Fig. 5.12). It appears normal which rules out plain obstruction as a possible reason. But she knows that the flank pain in combination with sulfadiazine can be a clear explanation for the kidney trouble that the patient is reporting. So she immediately stops administration of sulfadiazine and gives hydration instead. Following this treatment, the patient soon returns to normal. The eye also recovers as a result of the vitrectomy taken at visit 3.

5.6 EVALUATION

To evaluate the usability and efficacy of our prototype we interfaced it with an EMR database at a large teaching hospital. We invited six physicians (some were residents) and two health informatics (HI) professionals to participate in a pilot user study. None had previous experience with our system. All physicians were familiar with their current EMR systems, and the two HI professionals had much experience in designing and developing HI systems.

We first gave each participant a six-minute tutorial about our system. We explained the idea behind each layout and the basic functionalities, including the search and filter facilities, the three different interaction modes, the body map, etc. We prepared two sets of questions. The first set aimed at finding out whether our system can help physicians to quickly and accurately find information. The second set was more focused on design details along with some general questions. All six physicians did both sets of questions while the HI professionals were only given the second set. Our study was conducted with a set of real patients from the hospital EMR database.

5.6.1 QUESTIONS

The first set contained three questions, which were designed to test the efficacy of our system in terms of understanding the patient history and obtaining diagnostic assistance. All three

questions had fully defined answers so we could test accuracy. We also recorded the time to find the correct answer.

Q1. What were the most severe diseases of the patient?

Q2. What were the anatomical locations of these diseases?

Q3. What were the symptoms and which are the related tests that had been prescribed?

The second set was designed to test the usability.

Q4. Did you find it hard to read text in the radial layout?

Q5. Was adding links manually in the box-layout helpful?

Q6. Did the collapse, expand, zoom, and rotate interactions in the radial layout affect your mental map?

Q7. Did the system save you time compared to standard systems (could be paper-based or EMR systems)?

Q8. On a scale of 1 – 10, how would you rate the system?

5.6.2 RESULTS

For question set 1, all six physicians correctly identified the most severe disease, the anatomical location of this disease and the related symptoms and tests. Thus the accuracy was 100%.

The time for answering Q1 was between 4-10s (mean 6.5s), for Q2 it was between 2-11s (mean 5.6s) and for Q3 it was between 3-9s (mean 5.7s). For Q2, one physician knew the disease, we did not count his rapid answer (<1s). Another physician took more than 10s, stating that since the system was new it took some time to "learn where everything was". But eventually all physicians felt very comfortable with the system.

We found if physicians had the choice between sequential or radial layout, they would prefer the former. For example, for Q1, four doctors tried to find the answer using the sequential layout while two used the radial layout. Five used the severity filter to highlight the most severe diseases while one simply turned on the severity bar and found the longest one.

All physicians except the one who is very familiar with the disease used the body map in the radial layout to answer Q2. Five physicians used the *browse* mode (Fig. 5.9) to highlight the related symtoms and tests and finished the task quickly. One did not use the *browse* mode and tried to search through all links. This physician spent more than twice the time to identify the relation.

From question set 1 we learned that while all physicians felt that this system was vastly different from what they were used to, they became comfortable with it quickly and could efficiently locate the information we asked them to find.

Question set 2, on the other hand, was not specific to a certain medical case. It was designed

to get directions of further system improvements.

For Q4, we were interested in finding out whether the non-horizontal text in the radial layout was hard to read. One physician indeed wished we could keep the text horizontal, while another physician and one HI professional also noted that it was hard but that it was the natural way of displaying text on a radial layout, and that "your eyes will adjust to it". All others found it to be no problem.

For Q5, three physicians said they would gladly spend time on adding the links because it would save them much time when telling the next physician or the patient what was going on. Two physicians said they were not sure about the usefulness of the links, but they would add them. One physician said that he probably would not use this function. Finally, the two HI professionals said that this was an interesting function and they liked it, but since they were not physicians they could not tell under what circumstances they would use it.

For Q6, the two HI professionals thought the interactions provided a good way to let them focus on what they thought was interesting. Three of the physicians said that it would not affect the mental map because it was easy to figure out where everything was. Two physicians thought the same node should always stay at the same location, and one was not sure. But when we asked the latter three whether it was necessary to remove these interactions, they all said that they would like to keep them, and mentioned that although they did not use them when learning about the patient history, it could be helpful when making a report because they would have control over where to put the important nodes.

For Q7, four physicians thought our system would definitely save them time compared to the traditional EMR system they used. The other two were not sure, stating that they were very familiar with the current system and were satisfied with it. One of them said that a "well written paper report would actually save your time" and that she "already got used to it". But she admitted that very few reports were actually well written. The two HI professionals liked the system exceedingly well.

For Q8, three subjects said they would not be able to give a score unless they used the system intensively. The remaining five scores are: 5; 6; 7; 9 but has the potential to get 10; 7 for the sequential and 9 for the radial layout.

We find these evaluation results to be very encouraging, especially the fact that after a short tutorial most of the participants were comfortable with our system. We also gathered many valuable suggestions. One was that in case of large data, to reduce clutter, we could have some 'pre-filters' that would first pull records within some time frame, to let physicians retrieve the information that they think is important or interesting and then work on only what is left. Another suggestion was to reduce the size of the body map in the radial display; the body map did not need to be high resolution since doctors were familiar with the location references. This way we could save space for the outer rings to make the texts more readable. Also, among all features, the one that the physicians liked the most was the browsing highlight mode (see Fig. 5.9). They mentioned that "this feature can really help to explain what was going on with the patient".

5.6.3 CODING SUPPORT

We also demonstrated our system to a group of 6 medical coders employed at our institution. The group leader has been working on coding for more than 15 years. All the others have more than 3 years of coding experience. They were very positive and said that the system could save tremendously in time, 15s to 1 minute for each code lookup. Usually each coder deals with about 80-100 codes per day, so a rough estimation of the time that can be saved by using the system is 30 minutes to 1.5 hours for each person. Furthermore, they mentioned that the system would lead to a much more accurate coding and so reduce the time required for insurance claims to go through. Finally, they praised our hierarchical radial interface as an excellent platform for training, as the hierarchies are visually well presented and fully interactive—as opposed to the large books that are presently in use.

5.7 DISCUSSION

Presently, our system enables doctors to (1) quickly browse a patient's medical history via both the radial and sequential interface, and (2) enter new medical information using various input widgets in the sequential interface. For example, doctors can enter the name/description of a medical facet either via a textbox with full typing support or via searchable and scrollable lists of terms associated with standardized medical codes. This input interface is similar to that used in current commercial EMR systems.

We acknowledge, however, that while standardized medical codes do carry a great deal of information, they are not rich enough to capture all possible medical findings. During the exam, it may have been determined that the heart beat was normal, a tumor was benign, or the blood pressure was only slightly elevated. This information can be very valuable for future diagnoses and it is why doctors always resort to the patient's medical reports to get the full picture. Extending our system beyond the constraints of medical billing codes is a current focus of our work. Our first step in that direction was to provide doctors with convenient interfaces for adding links and severity levels. Other information can be accessed with a history browser which has hot links to the actual medical report(s) or image(s) (Fig. 5.12) associated with the corresponding node.

Chapter 6

CONCLUSION AND FUTURE WORK

We have presented an interactive network-based interface coupled with a parallel coordinates display, offering users various interaction tools to control the dimension ordering and assess correlation and causation relationships in the data. The framework can serve both as a data exploration environment and as an interactive presentation platform to demonstrate, explain, and justify identified relationships. We showed that the synergy of the network and parallel coordinate display offers a great deal of analytical power to users and it also allows them to explore their datasets more efficiently. Due to its resemblance to popular routing tools used in online maps the interaction with our system is also intuitive and natural.

We then utilized the interactive framework to visualize and analyze multi-field geospatial data in a dual-domain framework consisting of a geographic and a multivariate visualization interface. We used parallel coordinates for the latter and proposed a few enhancements both for interaction and for visualization. But our main focus was put on the interactions in and with the geo-display. Here we have taken advantage of Google Earth as a versatile 3D geo-browser. We showed how this platform can be programmed to enable in-window selection and brushing operations and how these can be coupled with the parallel coordinate display. We also showed how some multivariate information can be directly embedded into Google Earth, in form of pie and bar chart design primitives. We believe that our greatest contribution is the fact that our system was conceived in tight collaboration with a team of domain scientists and already enabled them to make a number of groundbreaking discoveries in their research field – the effect of aerosols on global warming. Our system is able to manage large heterogeneous data collections without problem and makes it easy to associate spatial effects in multi-field geo-referenced data.

We also presented the Five W's scheme of multivariate data gathering and reporting, with a special application to health care informatics. We have shown and evaluated that our framework can significantly lower the time and effort needed to access the patient's medical information, which is essential to arrive at a diagnostic conclusion. Finally, it was interesting to see that our system could also be helpful to medical coders.

6.1 FUTURE WORK

One major application of our framework is to do visual correlation analysis. A present limitation is that correlation can show only pairwise relationship of two single variables, but strong relationships may exist between two sets of variables. For example, while area (=width*length) could be correlated with price, neither width nor length might be. A possible solution can be to use regression and subspace analysis [65], which is area we would like to work on more. Subspace analysis would create islands in the landscape. However, a potential difficulty is that dimensions may appear in more than one subspace, which is not fully captured in this isolated island paradigm. Moreover, every correlation model for numerical variables assumes some specific data distribution. In this article, we addressed the most common one—the normal distribution. Since Pearson's correlation is only defined for this distribution, there may be other correlation metrics that can be applicable for others. Finally, correlation can be affected by outliers, non-linear relationships, heteroskedasticity, and multicollinearity. To gain more statistical robustness, it requires techniques of outlier detection and/or removal, and methodologies for detecting and visualizing non-linear relationships and relationships among multiple variables. We would like to investigate solutions to these statistical problems.

As we discussed in Section 3.9, statistical models can sometimes be represented by graphs or networks: random variables are nodes, and relationships of direct statistical dependence are shown as edges. This is called the graphical models. However, one big challenge in the model theories is the model discovery—identifying the structure of the network (graph). Given N variables in a dataset, there are $N!$ possible different structures of the network. Thus, it is infeasible to explore all possible model settings. As a result, efficient tools are needed to suggest a manageable subset of controllable variables (parameters). We have shown how our framework can help analysts on model discovery for causation analysis. We would like to extend our framework to assist in more model discovery applications. For example in control theory, given a complex system (plant), the goal of control theory is to develop techniques for the semi-automatic synthesis of a controller, which requires the existence of models of the plant. There are two types of models: Single-Input-Single-Output (SISO) model and Multiple-Input-Multiple-Output (MIMO) model. A SISO model has only one input parameter and one output parameter; while a MIMO model could have multiple input parameters and multiple output parameters. Models are learned from an approximation of the input and output behaviors of the plant. The choice of the input parameters is very important: they should expose the associations with the outputs. However, various parameters and associations are buried in various configuration scripts. Finding these parameters and associations from the various inputs and measured outputs poses challenges to analysts, engineers, and researchers alike. For parameter selection, unfortunately, the plants have a large number of compile-time and run-time parameters, which poses challenges to model identification. Take the SISO model, for example. Suppose there are N_I measured inputs and N_O measured outputs; in the worst case, the users need to try every possible combination of the input variables and output variables (which is N_I*N_O possible cases) for model identification. The MIMO model could result in even more possible cases. Thus, it is infeasible to explore all possible model settings. As a result, efficient tools are needed to suggest a manageable subset of controllable parameters. Depending on the applications, there are different rules and metrics to guide the parameter selection. Some applications are focusing on

correlation, then we can make an easy extension from our framework; some are focusing on other metrics, then we need to adjust our framework to properly encode these different metrics. We would like to make our framework more configurable and flexible in order to make it suitable for a wider variety of model discovery scenarios.

BIBLIOGRAPHY

[1] J. Abello, and K. Jeffrey. MGV: A system for visualizing massive multidigraphs. *IEEE Transactions on Visualization and Computer Graphics*, 8.1 (2002): 21-38.

[2] W. Aigner, S. Miksch, Supporting Protocol-Based Care in Medicine via Multiple Coordinated Views, *Proc. Coordinated and Multiple Views in Exploratory Visualization*, pp. 118-129, 2004.

[3] W. Aigner, S. Miksch, W. Müller, H. Schumann, C. Tominski, Visual Methods for Analyzing Time-Oriented Data, *IEEE Trans. on Visualization and Computer Graphics,* 14(1):47-60, 2008.

[4] B. A. Albrecht, Aerosols, Cloud Microphysics, and Fractional Cloudiness. *Science,*1989,245(4923),1227-1230.

[5] Alpern, L. Carter, The Hyperbox, *Proc. IEEE Visualization*, pp. 133–139, 1991.

[6] G. Andrienko, N. Andrienko, Exploring spatial data with dominant attribute map and parallel coordinates, *Env. and Urban Systems*, 2001, 25(1), 5-15.

[7] M. Ankerst, S. Berchtold, D. Keim, Similarity clustering of dimensions for an enhanced visualization of multidimensional data,*Proc. IEEE InfoVis,* pp. 52-60, 1998.

[8] M. Ankerst, D. A. Keim, and H. Kriegel. *Circle segments: A technique for visually exploring large multidimensional data sets*. Bibliothek der Universität Konstanz, 1996.

[9] F. Anscombe, Graphs in Statistical Analysis. *American Statistician*, 27 (1): 17–21, 1973.

[10] K. Arakawa, S. Tamaki, N. Kono, N. Kido, K. Ikegami, R. Ogawa, M. Tomita, Genome Projector: zoomable genome map with multiple views, *BMC Bioinformatics,* 10 (31), 2009.

[11] A. Artero, M. de Olivera, H. Levkowitz, Enhanced high-dimensional data visualization through dimension reduction and attribute arrangement, *Proc. IEEE InfoVis*, pp. 707–712, 2006.

[12] R. Bade, S. Schlechtweg, S. Miksch, Connecting Time-oriented Data and Information to a Coherent Interactive Visualization, *Proc. CHI,* pp. 105-112, 2004.

[13] L. A. Barrie, Arctic air pollution: An overview of current knowledge. *Atmospheric Environment - Part A General Topics,* 1986, 20(4), 643-663.

[14] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

[15] L. Best, A. Hunter and B. Stewart. Perceiving relationships: a physiological examination of the perception of scatterplots, *D. Barker-Plummer et al. (Eds.): Diagrams 2006*, pp. 244-257, 2006.

[16] G. Bigg, E. Rohling, An oxygen isotope data set for marine water, *J. Geoph. Res*., 2000, 105 (C4) 8527-8535.

[17] J. Blaas, C. Botha, F. Post, Interactive visualization of multi-field medical data using linked physical and feature-space views, *EuroVis*, 2007, 123-130.

[18] I. Blanco, A. Vega, A. González. Correlation visualization of high dimensional data using topographic maps, *International Conf. on Artificial Neural Networks*. 2415(134). 2002.

[19] C. Brewer, Color use guidelines for data representation, *Proc. Section on Statistical Graphics,* pp. 55-60, 1999.

[20] D. Brodbeck, L. Girardin. Visualization of large-scale customer satisfaction surveys using a parallel coordinate tree, *Proc. IEEE InfoVis* pp. 197-201, 2003.

[21] F. Chapman, *Winforms-geplugin-controls-library. http://code.google.com/p/winforms-geplugin-control-library/.* 2011.

[22] C. Chen, H. Hwu, W. Jang, C .Kao, Y. Tien, S. Tzeng, H. Wu. Matrix visualization and information mining, *Proc. Computational Statistics*, pp. 85-100, 2004

[23] C. Chen, C. Wang, K. L. Ma, A. Wittenberg. Static correlation visualization for large time-varying volume data, *Proc. IEEE Pacific Visualization*, pp. 27-34. 2011.

[24] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association* 68.342 (1973): 361-368.

[25] J. Claessen, J. van Wijk, Flexible linked axes for multivariate data visualization, *IEEE Trans. on Visualization and Computer Graphics*, 17(12): 2310-2316, 2011.

[26] W. Cleveland, *Dynamic Graphics for Statistics*, Springer, 1988.

[27] J. Cohen. The cost of dichotomization. *Applied Psychological Measurement*, 7, pp. 249–253.1983.

[28] J. Cohen, P. Cohen, S. West, L. Aiken. *Applied Multiple Regression/Correlation Analysis for the Behavioral Science (3rd ed.)*. Routledge Academic, 2002.

[29] C. Collins, S. Carpendale, G. Penn, Docuburst: Visualizing document content using language structure, *Proc. EuroVis*, pp.1039–1046, 2009.

[30] J. Conway, N. Sloane, *Sphere Packings, Lattices and Groups – 3rd Edition,* Springer Verlag , 1998.

[31] W. Dansgaard, Stable Isotopes in Precipitation, *Tellus*, 1964, 16 (4) 436-468.

[32] A. Dasgupta, R. Kosara, Pargnostics: screen-space metrics for Parallel Coordinates, *IEEE TVCG*, 16(6):1017-1026, 2006.

[33] S. Diehl, F. Beck, M. Burch, Uncovering strengths and weaknesses of radial visualizations—an empirical approach, *IEEE Trans. on Visualization and Computer Graphics*, 16(6):935-942, 2010.

[34] N. Elmqvist, J. Fekete, Hierarchical aggregation for information visualization: overview, techniques, and design guidelines, *IEEE Trans. on Visualization and Computer Graphics*, 16(3):439-454, 2010.

[35] N. Elmqvist, P. Dragicevic, J.-D. Fekete, Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation, *IEEE TVCG*, 14(6):1141–1148, 2008.

[36] A. Faiola, S. Hillier, Multivariate relational visualization of complex clinical datasets in a critical care setting: a data visualization interactive prototype, *Proc. Information Visualization,* pp. 460-468, 2006.

[37] U. Fayyad, G. G. Grinstein and A. Wierse, *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann Publishers, San Francisco, 2002.

[38] B. Ferdosi. J. Roerdink, Visualizing High-Dimensional Structures by Dimension Ordering and Filtering using Subspace Analysis, C*omputer Graphics Forum*, 30(3):1121–1130, 2011.

[39] V. Ferronsky, V. Brezgunov, Stable isotopes and ccean dynamics, *Environmental Isotopes in the Hydroshpere*. John Wiley, New York, 1982, 1-27.

[40] J. Fox and R. Thomson., Decision Support and Disease Management: A Logic Engineering Approach, *IEEE Transactions on Information Technology in Biomedicine*, 2(4):217–228, 1998.

[41] M. Friendly. Mosaic displays for multi-way contingency tables. *Journal of the American Statistical Association* 89.425 (1994): 190-200.

[42] M. Friendly. Extending mosaic displays: Marginal, conditional, and partial views of categorical data. *Journal of Computational and graphical Statistics* 8.3 (1999): 373-395.

[43] Y. Fua, M. Ward. E. Rundensteiner, Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces, *IEEE TVCG. 6*(2):150–159, 2000.

[44] G. W. Furnas. Generalized fisheye views. In *Proceedings of ACM CHI'86 Conference on Human Factors in Computing Systems*, pp. 16-23, Boston, Massachusetts, 1986.

[45] L. Gosink, C. Garth, J. Anderson, W. Bethel, K. Joy, An application of multivariate statistical analysis for query-driven visualization, *IEEE Trans. Vis. Comput. Graph*. 2011, 17(3) 264-275.

[46] M. Graham, J. Kennedy, Using curves to enhance parallel coordinate visualizations, *Proc. IEEE Info Vis*. pp.10-16, 2003.

[47] M. Ghoniem, J.-D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114-135, 2005.

[48] D. Guo, J. Chen, A. MacEachren, K. Liao, A visualization system for space-time and multivariate patterns, *IEEE Trans. Vis. Comput. Graph*. 2006,12(6):1461-1474.

[49] H. Guo, H. Xiao, and X. Yuan. Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates. *Pacific Visualization Symposium (PacificVis)*, IEEE, 2011.

[50] D. Hadorn, Use of Algorithms in Clinical Practice Guideline Development: Methodology Perspectives *AHCPR Pub.*, 9(95):93–104, 1995.

[51] K. Hara, et al. Mixing states of individual aerosol particles in spring Arctic troposphere during ASTAR 2000 campaign. *J. of Geophysical Research-Atmospheres.* 2003.

[52] J. Hartigan. Direct clustering of a data matrix, *Journal of the American Statistical Association*, 67 (337):123-129. 1972.

[53] J. Hartigan, Printer graphics for clustering, *Journal of Statistical Computation and Simulation,* 4(3):187-213, 1975.

[54] M. A. Hearst. TileBars: visualization of term distribution information in full text information access. *Proceedings of the SIGCHI conference on Human factors in computing system*s. ACM Press/Addison-Wesley Publishing Co., 1995.

[55] J. Heer, M. Bostock, V. Ogievetsky, A tour through the visualization zoo, *Commun. ACM*, 53(6):59-67, 2010.

[56] N. Henry, J.-D. Fekete. MatrixExplorer: a dual-representation system to explore social networks, *IEEE Trans. on Visualization and Computer Graphics*, 12 (5), 2006.

[57] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, E. Stanley, DNA visual and analytic data mining., *Proc. IEEE Visualization.*, pp. 437-441. 1997.

[58] D. Holten, Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Trans. Vis. and Computer Graphics*, 12(5):741-748, 2006.

[59] W. Horn, C. Popow, L. Unterasinger, Support for fast comprehension of ICU data: visualization using metaphor graphics, *Methods of Information in Medicine,* 40:421-424, 2001.

[60] A. Inselberg, B. Dimsdale, Parallel coordinates: A tool for visualizing multi-dimensional geometry, *Proc. IEEE Visualization,* pp. 361-378, 1990.

[61] A. Inselberg, T. Avidan, The automated multidimensional detective, *Proc. IEEE InfoVis*, pp. 112-119, 1999.

[62] M. Jern, T. Astrom, S. Johansson, GeoAnalytics tools applied to large geospatial datasets, *Proc. IEEE Information Visualization,* 2008, 362-372.

[63] S. Johansson, M. Jern, J. Johansson. Interactive quantification of categorical variables in mixed data sets, *Proc. Conf. on Information Visualization,* pp. 3-10. 2008.

[64] S. Johansson, J. Johansson, Interactive dimensionality reduction through user-defined combinations of quality metrics, *IEEE TVCG. 15*(6):993–1000, 2009.

[65] B. Johnson, B. Shneiderman, Tree maps: A space-filling approach to the visualization of hierarchical information structures, *Proc. IEEE Visualization*, pp. 284-291, 1991.

[66] E. Kandogan. Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. Proceedings of the *IEEE Information Visualization Symposium*. Vol. 650. 2000.

[67] D. Keim, F. Mansmann, J. Schneidewind, et al. (2008). Visual analytics: Scope and challenges. *Visual Data Mining*, 76-90.

[68] D. Keim, Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 7(2), 100–107. 2002.

[69] D. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, 6.1 (2000): 59-78.

[70] D. Keim, M. Ankerst, and H. Kriegel. Recursive pattern: A technique for visualizing very large amounts of data. *IEEE 6th conference on Visualization'95*. 1995.

[71] D. Keim, and H-P. Kriegel. VisDB: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications,* 14.5 (1994): 40-49.

[72] S. Kisilevich, D. Keim, L. Rokach, GEO-SPADE: a generic Google Earth-based framework for analyzing and exploring spatio-temporal data, *Int' Conference on Enterprise Info. Systems*, 2010, 13-20.

[73] A. Klippel, F. Hardisty, R. Li, and C. Weaver. Colour-Enhanced Star Plot Glyphs: Can Salient Shape Characteristics Be Overcome?. Cartographica: *The International Journal for Geographic Information and Geovisualization*, 44(3), 217-231. 2009.

[74] R. Kosara, S. Miksch, Visualization Techniques for Time-Oriented, Skeletal Plans in Medical Therapy Planning, Proc. Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making, pp. 291-300, 1999.

[75] E. Koua, A. Maceachren, M. Kraak, Evaluating the usability of visualization methods in an exploratory geo-visualization environment. *J. of Geographical Information Science*, 2006, 20(4) 425-448.

[76] H. Kriegel, P. Kröger, A. Zimek, Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering, *ACM Trans KDD*, 3(1), 2009.

[77] J. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, *Psychometrika,* 29(1):1–27, 1964.

[78] L. Kühne, J. Giesen, Z. Zhang, S. Ha, K. Mueller. Data-Driven Approach to Hue-Preserving Color-Blending. IEEE Transactions on Visualization and Computer Graphics. 18(12):2122-2131. 2012.

[79] G. Langelier, H. Sahraoui, P. Poulin, Visualization-based analysis of quality for large-scale software systems, *Proc. IEEE/ACM Automated Software Engineering*, pp. 214-223, 2005.

[80] H. Levkowitz. Color icons-merging color and texture perception for integrated visualization of multiple parameters. Proceedings., *IEEE Conference on Visualization*, 1991.

[81] A. Lex, M. Streit, C. Partl, K, Kashofer, D. Schmalstieg. Comparative Analysis of Multidimensional, Quantitative Data. *IEEE TVCG*, 16(6): 1027-1035. 2010.

[82] J. Li, J.-B. Martens, J. van Wijk, Judging correlation from scatterplots and parallel coordinate plots, *Information Visualization*, 9(1):13–30, 2010.

[83] L. Lins, M. Heilbrun, J. Freire, C. Silva. VisCareTrails: Visualizing Trails in the Electronic Health Record with Timed Word Trees, a Pancreas Cancer Use Case. *IEEE VAHC Workshop*, 2011.

[84] K. N. Liou, and S. C. Ou, The Role of Cloud Microphysical Processes in Climate - an Assessment from a One-Dimensional Perspective. *J. Geophys. Res.-Atmos.*1989, 94, 8599-8607.

[85] S. Ma, J. Hellerstein. Ordering categorical data to improve visualization, *Proc. IEEE Information Visualization*, pp. 15-17, 1999.

[86] A. MacEachren, M. Wachowicz, R. Edsall, D. Haug, R. Masters, Constructing knowledge from multivariate spatiotemporal data: integrating geographical visualization with knowledge discovery in database methods, *Int' J. of Geographical Information Science,* 1999, 13(4) 311-334.

[87] K. McDonnell, K. Mueller, Illustrative Parallel Coordinates, *Computer Graphics Forum*, 27(3):1031-1027, 2008.

[88] G. McFarquhar et al.: Indirect and semi-direct aerosol campaign (ISDAC): the impact of arctic aerosols on clouds, *Bulletin of the American Meteorological Society,* 2011, 92(2) 183-201.

[89] G. McFiggans, P. Artaxo, U. Baltensperger, et al. The effect of physical and chemical aerosol properties on warm cloud droplet activation. *Atmospheric Chemistry and Physics,* 2006, 6(9), 2593-2649.

[90] D. Meyer, Z. Achim, and H. Kurt. Visualizing independence using extended association plots. *Proceedings of DSC* 2003 (2003).

[91] M. Meyer, H. Lee, A. Barr, M. Desbrun, Generalized Barycentric Coordinates on Irregular Polygons, *Graphics Tools*, pp. 1086-7651, 2002.

[92] W. Mook, The Oxygen-18 content of rivers, *SCOPE*, 1982, 52 (1982) 565-570.

[93] S. Nagaraj, V. Natarajan, Relation-aware isosurface extraction in multifield data, *IEEE Trans. Vis. Comput. Graph.*, 2011, 17(2) 182-191.

[94] E. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre, ClusterSculptor: A Visual Analytics tool for high-dimensional data, *IEEE Symposium on Visual Analytics Science and Technology* , 2007, pp. 75-82.

[95] J. Nam, K. Mueller, TripAdvisorN-D: A Tourism-Inspired High-Dimensional Space Exploration Framework with Overview and Detail, *IEEE Trans. Visualization and Computer Graphics*, 19(2): 291-305, 2013.

[96] E. Niklas, J. Stasko, and P. Tsigas. DataMeadow: a visual canvas for analysis of large-scale multivariate data. *Information visualization* 7.1 (2008): 18-33.

[97] C. North, B. Shneiderman, Snap-together visualization: Can users construct and operate coordinated visualizations? *Int. J. Human.-Computer Studies.* 53(5):715-739, 2000.

[98] P. Ordonez, M. desJardins, M. Lombardi, C. Lehmann, J. Fackler,An animated multivariate visualization for physiological and clinical data, *Proc. ACM Int. Health Informatics Symposium*, pp. 771-779, 2010.

[99] J. Pearl. *Causality: models, reasoning and inference.* Vol. 29. Cambridge: MIT press, 2000.

[100] J. Pearl and T. Verma. A Theory of Inferred Causation. *Proc. of the Second Int. Conf. on Knowledge Representation and Reasoning.* pp. 441–452, 1991.

[101] J. P. Pellet, A. Elisseeff. A partial correlation-based algorithm for causal structure discovery with continuous variables. *Advances in Intelligent Data Analysis VII*. Springer Berlin Heidelberg. pp. 229-239. 2007.

[102] W. Peng, M. Ward, E. Rundensteiner, Clutter reduction in multi-dimensional data visualization using dimension reordering, *Proc. IEEE InfoVis*, pp. 89–96, 2004.

[103] R. Pickett, and G. Georges. Iconographic displays for visualizing multidimensional data. *Proc. IEEE Conf. on Systems, Man and Cybernetics*, IEEE Press, Piscataway, NJ. Vol. 514. 1988.

[104] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, B. Shneiderman, Lifelines: Using visualization to enhance navigation and analysis of patient records, *Proc. AMIA Annual Symposium*, pp. 76–80, 1998.

[105] S. Powsner, E. Tufte, Graphical summary of patient status, *The Lancet*, 344(8919):386-389, 1994.

[106] H. Qu, W. Chan, A. Xu, K. Chung, K. Lau, P. Guo, Visual analysis of the air pollution problem in Hong Kong, *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1408-1415, 2007.

[107] S. Quaglini, M. Stefanelli, G. Lanzola, V. Caporusso, S. Panzarasa, Flexible guideline-based patient careflow systems, *Artificial Intelligence in Medicine*, 22(1):65–80, 2001.

[108] P. K. Quinn, T. S. Bates, E. Baum, et al. Short-lived pollutants in the Arctic: their climate impact and possible mitigation strategies. *Atmospheric Chemistry and Physics* 2008, 1723-1735.

[109] L. F. Radke, J. H. Lyons, D. A. Hegg, P. V. Hobbs, and I. H. Bailey, Airborne Observations of Arctic Aerosols .1. *Characteristics of Arctic Haze. Geophysical Research Letters,* 1984, 11. 393-396.

[110] A. Rind, S. Miksch, W. Aigner, T. Turic, M. Pohl, VisuExplore: Gaining new medical insights from visual exploration*, Proc. Int. Workshop Interactive Systems in Healthcare,* pp. 149-152. 2010.

[111] T. Ropinski, I. Viola, M. Bierman, H. Hauser, K. Hinrichs, Multimodal visualization with interactive closeups, *EGUK Theory and Practice of Computer Graphics (TPCG)*, 2009.

[112] G. E. Rosario, E. A. Rundensteiner, D. C. Brown, and M. O. Ward, Mapping Nominal Values to Numbers for Effective Visualization, *Information Visualization*, 3(2): 80-95, 2004.

[113] P. Royston, D. Altman, W. Sauerbrei. Dichotomizing continuous predictors in multiple regression: a bad idea, *Stat Med*, 25:127-141, 2006.

[114] G. Savov, J. Masanz, P. Ogren, et al. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications, *J Am Med Inform Assoc*., 17(5):507-13, 2010.

[115] E. Segel, J. Heer, Narrative visualization: Telling stories with data, *IEEE Trans. Vis. and Comp. Graphics*, 16(6):1139-1148, 2010.

[116] Y. Shahar, D. Goren-bar, D. Boaz, G. Tahan, Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions *J. Artificial Intelligence in Medicine*, 38(2):115-135, 2006.

[117] G.A., Schmidt, G. R. Bigg and E. J. Rohling. Global Seawater Oxygen-18 Database - v1.21 *http://data.giss.nasa.gov/o18data/* . 1999.

[118] G. Schmidt, G. Bigg, E. Rohling, Global Seawater Oxygen-18 Database - v1.21 (accessed 7/10/11). *http://data.giss.nasa.gov/o18data.*

[119] C. Schmid, H. Hinterberger, Comparative multivariate visualization across conceptually different graphic displays. *Proc. SSDBM*, pp. 42-51, 1994.

[120] J. Seo, B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data, *Information Visualization*, 4(2): 96-113, 2005.

[121] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations *Proc. IEEE Visual Languages*, pp. 336 -343. 1996.

[122] H. Siirtola. Direct manipulation of parallel coordinates, *Proc. IEEE Information Visualization*, pp. 373–378, 2000.

[123] H. Siirtola, E. Mäkinen. Constructing and reconstructing the reorderable matrix. *Information Visualization*, 4(1):32-48. 2005.

[124] A. Sirois, L. A. Barrie, Arctic lower tropospheric aerosol trends and composition at Alert, Canada: 1980-1995. *Journal of Geophysical Research-Atmospheres,* 1999, 104, 11599-11618.

[125] J. Stasko, E. Zhang, Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations, *IEEE Symp. on Information Visualization,* pp. 57-65, 2000.

[126] J. Sukharev, C. Wang, K. L. Ma, A. Wittenberg. Correlation study of time-varying multivariate climate data sets, *IEEE PacificVis,* pp. 161-168, 2009.

[127] A. Tatu, G. Albuquerque, M. Eisemann, H. Theisel, M. Magnor, D. Keim, Combining automated analysis and visualization techniques for effective exploration of high-dimensional data, *Proc. IEEE VAST*, pp. 59–66, 2009.

[128] H. Theisel. Higher order Parallel Coordinates, *Proc. VMV*, pp. 415-420, 2000.

[129] Theus, Martin. Interactive data visualization using mondrian. *Journal of Statistical Software* 7.11 (2003): 1-9.

[130] W. Torgerson, Multidimensional scaling: I. theory and method, *Psychometrika*, 17:401–419, 1952.

[131] C. Turkay, P. Filzmoser, H. Hauser. Brushing dimensions - a dual visual analysis model for high-dimensional data. *IEEE Trans. on Visualization and Computer Graphics*, 17(12): 2591–2599, 2011.

[132] H. Wainer. Finding what is not there through the unfortunate binning of results: The Mendel effect, *Chance*, 19(1):49-56, 2006.

[133] M. Ward, XmdvTool: Integrating multiple methods for visualizing multivariate data, *Proc. IEEE Visualization.*, pp. 326-333, 1994.

[134] C. Weaver, Cross-Filtered Views for Multidimensional Visual Analysis, *IEEE Trans. Visualization and Computer Graphics*, 16(2):192-204, 2010.

[135] S. West, L. Aiken, J. Krull. Experimental personality designs: Analyzing categorical by continuous variable interactions, *Journal of Personality*, 64, 1–49. 1996.

[136] K. Wongsuphasawat and D. Gotz. Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization. *IEEE Trans. Visualization and Computer Graphics*, 18(12): 2659- 2668, 2012.

[137] M. Workman, M. F. Lesser, J. Kim, An exploratory study of cognitive load in diagnosing patient conditions, *Int. Journal for Quality in Health Care*, 19(3):127-133, 2007.

[138] L. Young, H. Bazari, M. Durand, J. Branda, Case 33-2010 — A 22-Year-Old Woman with Blurred Vision and Renal Failure, *New England J Medicine,* 363:1749-1758, 2010.

[139] P. Wong, R. Bergeron, Multivariate visualization using metric scaling, *Proc. IEEE Visualization*, pp. 111-118, 1997.

[140] J. Wood, J. Dykes, A. Slingsby, K. Clarke, Interactive visual exploration of a large spatio-temporal dataset: reflections on a geovisualization mashup, *IEEE TVCG*. 2007, 13(6) 1176-1183.

[141] J. Yang, W. Peng, M. Ward, E. Rundensteiner, Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets, *Proc. IEEE InfoVis*, pp. 105-112, 2003.

[142] J. Yang, D. Hubball, M. Ward, E. Rundensteiner, W. Ribarsky. Value and Relation Display: Interactive visual exploration of large data sets with hundreds of dimensions, *IEEE Trans. on Visualization and Computer Graphics*, 13(3):494 –507, 2007.

[143] J. Yang, M. Ward, E. Rundensteiner, S. Huang, Visual hierarchical dimension reduction for exploration of high dimensional datasets, *Proc. VisSym*, pp. 19–28, 2003.

[144] Yi, J. S., Melton, R., Stasko, J., & Jacko, J. A. (2005). Dust & Magnet: multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4), 239-256.

[145] Yi, J. S., ah Kang, Y., Stasko, J. T., & Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. *, IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1224-1231.

[146] X. Yuan, P. Guo, H. Xiao, H. Zhou, H. Qu, Scattering points in parallel coordinates, *IEEE TVCG*, 15(6):1001-1008, 2009.

[147] Z. Zhang, F. Ahmed, A. Mittal, IV. Ramakrishnan, R. Zhao, A Viccellio, K. Mueller, AnamneVis: A Framework for the Visualization of Patient History and Medical Diagnostics Chains, *IEEE VAHC Workshop*, 2011.

[148] Z. Zhang, D. Gotz, A. Perer. Interactive Visual Patient Cohort Analysis, *IEEE VAHC Workshop*, 2012.

[149] Z. Zhang, D. Gotz, A. Perer. Iterative Cohort Analysis and Exploration. *Information Visualization*. 2014.

[150] Z. Zhang, K. T. McDonnell, K. Mueller. A network-based interface for the exploration of high-dimensional data spaces. *Proc. IEEE Pacific Visualization*, pp. 17-24, 2012.

[151] Z. Zhang, A. Mittal, S. Garg, A.Dimitriyadi, IV Ramakrishnan, R. Zhao, A. Viccellio, K. Mueller, A Visual Analytics Framework for Emergency Room Clinical Encounters, *IEEE VAHC Workshop*, 2010.

[152] Z. Zhang, X. Tong, K. McDonnell, A. Zelenyuk, D. Imre, K. Mueller. An Interactive Visual Analytics Framework for Multi-Field Data in a Geo-Spatial Context. *Special Issue of Tsinghua Science and Technology on Visualization and Computer Graphics.* 18(2):111-124. April, 2013.

[153] A. Zelenyuk, D. Imre, Beyond single particle mass spectrometry: multidimensional characterisation of individual aerosol particles, *Int. Rev. Phys. Chem*. 2009, 28, 309–358.

[154] A. Zelenyuk, D. Imre, Z. Zhang, J. H. Lee, K. Mueller, K. McDonnell. An Interactive Visual Analytics Framework for Multidimensional Data in a Geo-Spatial Context. *American Association for Aerosol Research (AAAR) 32nd Annual Conference.* Portland, OR, Sept. 2013.

[155] A. Zelenyuk, J. Yang, D. Imre, E. Choi, SPLAT II: An aircraft compatible, ultra-sensitive, high precision instrument for in-situ characterization of the size & composition of fine & ultrafine particles. *Aerosol Sci. Technol*. 2009, 43, 411-424.

[156] Auto MPG dataset, http://archive.ics.uci.edu/ml/datasets/Auto+MPG

[157] Automobile dataset, http://archive.ics.uci.edu/ml/datasets/Automobile

[158] College Prowler (accessed 9/09), http://collegeprowler.com

[159] US News Best Colleges (accessed 9/09), http://colleges.usnews.rankingsandreviews.com

[160] *http://publicwiki.deltares.nl/display/OET/OpenEarth* (accessed 12/07/11)

[161] *http://publicwiki.deltares.nl/display/OET/KML+Screenshots* (accessed 12/07/11)

[162] FLARE: http://flare.prefuse.org/ (accessed on 9/10/2010)

[163] Defining and Testing EMR Usability. Healthcare Information and Management Systems Society (HIMSS), June 2009 (available at http://www.himss.org/content/files/himss_definngand testingemrusability.pdf, (accessed on 5/17/2012).

[164] http://lane.stanford.edu/portals/cvicu/HCP_Neuro_Tab_4/0-10_Pain_Scale.pdf (accessed on 5/15/2012)