

Dimension Reduction and Clustering for Interactive Visual Analytics

John E. Wenskovitch, Jr.

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science & Application

Christopher L. North, Chair

Nicholas F. Polys

Douglas A. Bowman

Scotland C. Leman

G. Elisabeta Marai

August 14, 2019

Blacksburg, Virginia

Keywords: Dimension Reduction, Clustering, Semantic Interaction, Visual Analytics

Copyright 2019, John E. Wenskovitch, Jr.

Dimension Reduction and Clustering for Interactive Visual Analytics

John E. Wenskovitch, Jr.

(ABSTRACT)

When exploring large, high-dimensional datasets, analysts often utilize two techniques for reducing the data to make exploration more tractable. The first technique, dimension reduction, reduces the high-dimensional dataset into a low-dimensional space while preserving high-dimensional structures. The second, clustering, groups similar observations while simultaneously separating dissimilar observations. Existing work presents a number of systems and approaches that utilize these techniques; however, these techniques can cooperate or conflict in unexpected ways.

The core contribution of this work is the systematic examination of the design space at the intersection of dimension reduction and clustering when building intelligent, interactive tools in visual analytics. I survey existing techniques for dimension reduction and clustering algorithms in visual analytics tools, and I explore the design space for creating projections and interactions that include dimension reduction and clustering algorithms in the same visual interface. Further, I implement and evaluate three prototype tools that implement specific points within this design space. Finally, I run a cognitive study to understand how analysts perform dimension reduction (spatialization) and clustering (grouping) operations. Contributions of this work include surveys of existing techniques, three interactive tools and usage cases demonstrating their utility, design decisions for implementing future tools, and a presentation of complex human organizational behaviors.

Dimension Reduction and Clustering for Interactive Visual Analytics

John E. Wenskovitch, Jr.

(GENERAL AUDIENCE ABSTRACT)

When an analyst is exploring a dataset, they seek to gain insight from the data. With data sets growing larger, analysts require techniques to help them reduce the size of the data while still maintaining its meaning. Two commonly-utilized techniques are dimension reduction and clustering. Dimension reduction seeks to eliminate unnecessary features from the data, reducing the number of columns to a smaller number. Clustering seeks to group similar objects together, reducing the number of rows to a smaller number.

The contribution of this work is to explore how dimension reduction and clustering are currently being used in interactive visual analytics systems, as well as to explore how they could be used to address challenges faced by analysts in the future. To do so, I survey existing techniques and explore the design space for creating visualizations that incorporate both types of computations. I look at methods by which an analyst could interact with those projections in order to communicate their interests to the system, thereby producing visualizations that better match the needs of the analyst. I develop and evaluate three tools that incorporate both dimension reduction and clustering in separate computational pipelines. Finally, I conduct a cognitive study to better understand how users think about these operations, in order to create guidelines for better systems in the future.

Acknowledgments

Who knew that writing acknowledgments would be the toughest part of finishing this dissertation? Grad school has certainly been a journey that lasted a few years longer than originally planned. I have several alphabetical lists of people who deserve acknowledgment; without them, I would have broken and given up long, long ago.

First, I have to thank family. It's immediately obvious that my work ethic came as a result of watching my parents over the last 31 years. They've always been hard-working and ready to take on responsibilities that they could have avoided or ignored. It's always in the back of my mind that I have a lot to live up to in order to best reflect the example that they set. Thanks, Mom and Dad.

Second, I have a number of mentors and colleagues to thank. Even beyond the decade of graduate school, a number of people have influenced my path over the years who deserve acknowledgment. Liz Marai and Chris North are certainly at the top of the list, having advised and mentored me through parts of this PhD. I can't discount the feedback and support that I've received from the other three members of my committee as well. Cal Ribbens, Greg Kapfhammer, and Larry Viehland have all been responsible for hiring me to continue to teach classes and interact with students almost daily, providing a great break from the occasional drudgery of research. Oliver Bonham-Carter, Janyl Jumadinova, Pauline Lanzine, Bob Roos, and David Wagner were all wonderful CS colleagues and support during my Allegheny intermission. A number of professors at Gannon University also influenced the direction that my life took, including but certainly not limited to Barry Brinkman, Michael Caulfield, John Coffman, Steve Frezza, Mei-Huei Tang, and Theresa Vitolo. And I can't skip Bob Campbell from all the way back in high school.

And so I guess that brings me around to the list of friends. I'm certain to forget a few names, so if you happen to be reading this, add yourself to the appropriate group. Michelle Dowling, Mike Himes, Tim Luciani, Kevin Mowry, Jim "Orlo" Overly, and Byron Rich have all been foundational cornerstones (yes, six corners, deal with it) to my life and post-graduate experiences. Around for a chat when needed, helping me unwind, chatting about something nerdy, and just generally being a surrogate family. Allison Bobby, Jean-Maurice DeMars, Lesley Fairman, Justin Furiga, Lata Kodali, Matt Slifko, and Nathan Wycoff have also been available to commiserate when needed. Along with Byron, Alice Allen, Sophia Brueckner, Amruta Jaodand, and Jamie Lombardi have helped me to escape the confines of computer science and stretch out my experiences to other fields (and led to me giving talks in seven extra countries). The AAAP and RVAS amateur astronomy organizations have supported my stargazing hobby extensively, and I've learned a lot from Frank Baratta, Dan Chrisman, Michael Good, Dwight Holland, Rowen Poole, Tom Reiland, and many more. Kristen Buccigrossi and Michelle McMeans were frequently around for support when I committed to running stupidly long distances.

Last but not least, this work was funded in part by NSF Grants IIS-1447416, IIS-1633363, and DGE-1545362, as well as by a grant from General Dynamics Mission Systems.

Attributions

The survey of dimension reduction algorithms in Chapter 2 and the survey of clustering algorithms in Chapter 3 were seeded in the Related Work section of “Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics,” published at VAST 2017 [320]. Coauthors on this work include Ian Crandell, Leanna House, Scotland Leman, Chris North, and Naren Ramakrishnan. Both of these sections were expanded and validated for mathematical accuracy with assistance from Nathan Wycoff.

The survey of dimension reduction tools in Chapter 2 was incrementally built up from a variety of publications. In addition to the aforementioned VAST 2017 paper, Michelle Dowling’s papers “SIRIUS: Dual, Symmetric, Interactive Dimension Reductions” [98] and “A Bidirectional Pipeline for Semantic Interaction” [97]. The version included in this dissertation is original work, but many of these tools were initially located while performing related work searches these papers. Other coauthors on these publications include Adam Binford, J.T. Fry, Peter Hauck, Leanna House, Scotland Leman, Chris North, and Nicholas Polys.

The survey of clustering tools in Chapter 3 was seeded by my submission to the Sensemaking in a Senseless World workshop at CHI 2018, “The Cognitive and Computational Benefits and Limitations of Clustering for Sensemaking” [321]. Coauthors on this work include Michelle Dowling and Chris North.

Chapter 4 consists of much of the remainder of the 2017 VAST paper and 2018 CHI workshop paper, with the same set of coauthors mentioned previously.

Chapter 5 was also written with the assistance of Michelle Dowling and Chris North. The full paper version will be revised and resubmitted to another conference. An abbreviated version was published as a poster at IUI 2019, “Simultaneous Interaction with Dimension Reduction and Clustering Projections” [323].

Chapters 6–9 were also coauthored by Chris North. Chapter 6 was published in the Human-in-the-Loop Data Analytics (HILDA) workshop at SIGMOD 2017 as “Observation-Level Interaction with Clustering and Dimension Reduction Algorithms” [317]. Chapter 7 has recently been accepted to Visualization in Data Science (VDS) 2019 as “Pollux: Interactive Cluster-First Projections of High-Dimensional Data” [318]. Chapters 8 and 9 are original material for this dissertation, and will be submitted to a conference or journal in the near future.

Chapter 10 has recently been accepted at the MLUI 2019 workshop on Machine Learning from User Interactions for Visualization and Analytics at IEEE VIS as “Machine Learning from User Interaction for Visualization and Analytics: A Workshop-Generated Research Agenda” [322]. Coauthors on this work are Remco Chang, Michelle Dowling, Alex Endert, Laura Grose, Chris North, and David H. Rogers.

Finally, this full collection of work, including both text and software, received many rounds of feedback from the members of the Virginia Tech Information Visualization (InfoVis) Lab and the Bayesian Visual Analytics (BaVA) group. Paragraphs and subsections throughout this dissertation were inspired, modified, and extended by conversations with members of these research groups.

Contents

- List of Figures** **xvii**

- List of Tables** **xxviii**

- 1 Introduction** **1**

- 2 Background: Dimension Reduction Algorithms and Tools** **7**
 - 2.1 Dimension Reduction Algorithms 8
 - 2.1.1 Linear Dimension Reduction 8
 - 2.1.2 Nonlinear Dimension Reduction 10
 - 2.1.3 Topic Modeling 15
 - 2.1.4 Feature Selection 16
 - 2.1.5 Feature Learning 17
 - 2.1.6 Sufficient Dimension Reduction 18
 - 2.1.7 Distance Functions 19
 - 2.2 Interactive Dimension Reduction Tools 21
 - 2.2.1 Quantitative Data 21
 - 2.2.2 Text Data 23
 - 2.2.3 Complex Data 24

2.2.4	Big Data	25
2.2.5	Immersive and 3D Spaces	25
2.2.6	Relationship to Clustering	26
3	Background: Clustering Algorithms and Tools	28
3.1	Clustering Algorithms	30
3.1.1	Hierarchical Clustering	30
3.1.2	Centroid Models	31
3.1.3	Distribution-Based Models	34
3.1.4	Density Models	35
3.1.5	Subspace Clustering	36
3.1.6	Evaluating Cluster Quality	37
3.1.7	Multiple Cluster Assignment	39
3.2	Interactive Clustering Tools	40
3.2.1	Interactive Clustering for Data Insight	41
3.2.2	Human-in-the-Loop Clustering	43
3.2.3	Getting Feedback	45
3.2.4	Studies	47
3.3	Future Work	48
4	Dimension Reduction and Clustering Projections	50

4.1	Tasks	51
4.1.1	Dimension Reduction and Clustering Tasks	54
4.2	Coordinating the Algorithms	58
4.2.1	Dimension Reduction and Clustering Combinations	59
4.3	Visual Representation	64
4.3.1	Visualization Challenges	68
4.3.2	Algorithm Order Visualizations	71
4.4	Discussion	74
4.5	Conclusion	76
5	Dimension Reduction and Clustering Interactions	78
5.1	Background	79
5.1.1	Current Interaction Techniques	80
5.1.2	Combined Interaction Techniques	83
5.1.3	“With Respect to What”	85
5.1.4	Pipelines in Visual Analytics	86
5.2	Motivating Example	88
5.3	Interactions on Observations	91
5.3.1	With Respect to Other Observations	92
5.3.2	With Respect to Clusters	95

5.4	Interactions on Clusters	100
5.4.1	Repositioning Interactions	100
5.4.2	Additional Cluster Interactions	103
5.5	Discussion	105
5.5.1	Visualizing the Feedback	105
5.5.2	Shared or Separate Weight Vectors	105
5.5.3	Parametric Interactions	106
5.5.4	Design Considerations	107
5.5.5	Towards Resolving Semantic Interaction Ambiguity	108
5.6	Conclusion	109
6	Castor: Dimension Reduction First	110
6.1	Development and Design	111
6.1.1	Development	111
6.1.2	Similarities and Differences Between the Tools	112
6.2	Model and Implementation	115
6.2.1	Projection Direction	116
6.2.2	Interaction Direction	118
6.3	Usage Scenario	123
6.4	Discussion	125

6.5	Limitations	125
6.6	Conclusion	126
7	Pollux: Clustering First	127
7.1	Pollux	128
7.1.1	Projection Direction	129
7.1.2	Interaction Direction	132
7.2	Extended Design Space	135
7.2.1	Edge Class Selection	135
7.2.2	The Effect of Edge Class Selection on Performance	136
7.2.3	The Role of Edge Class Weights	138
7.2.4	Alternate Visual Representations	139
7.2.5	Analyst Control of Cluster Count	140
7.2.6	Extending the Hierarchy	141
7.2.7	When to Learn?	142
7.2.8	Multiple Distance Functions and Weight Vectors	142
7.3	Evaluation	143
7.4	Discussion	146
7.4.1	Limitations and Future Work	148
7.5	Conclusion	149

8	Analyzing Pipeline Order Via Case Studies	150
8.1	Gemini	152
8.2	Insight Usage Scenario	154
8.2.1	Gemini	156
8.2.2	Castor	156
8.2.3	Pollux	157
8.2.4	Insight Trends	158
8.3	Quantitative Evaluation	158
8.3.1	Quality Metric Definitions	159
8.3.2	Quantitative Measures on Initial States	160
8.3.3	Quantitative Measures Across Interactions	163
8.4	Discussion	165
8.4.1	Tool Quality Summary	165
8.4.2	Design Considerations	166
8.4.3	Limitations and Future Work	167
8.5	Conclusion	168
9	Cognitive Dimension Reduction and Clustering	170
9.1	Background	171
9.2	Experimental Design	173

9.2.1	Participants	174
9.2.2	Dataset	174
9.2.3	Procedure	176
9.3	Organization Task Results	177
9.3.1	Beginning the Analysis	177
9.3.2	Cluster Structures	178
9.3.3	Cluster Operations	181
9.3.4	Decision Making	183
9.3.5	Timing	183
9.4	Update Task Results	184
9.4.1	Beginning the Analysis	184
9.4.2	Cluster Structures and Operations	186
9.4.3	Decision Making	186
9.4.4	Timing	187
9.5	Discussion	187
9.5.1	Overarching Strategies	187
9.5.2	Post-Survey	191
9.5.3	Complex Spaces	193
9.5.4	External Knowledge	194
9.5.5	“The Big Reveal”	196

9.5.6	Computational Aid for Complex Visualizations	196
9.5.7	Lessons for Future Tool Development	197
9.5.8	Limitations and Future Work	198
9.6	Conclusion	199
10	Human in the Loop Research Agenda	201
10.1	Workshop and Research Agenda Background	202
10.2	Interactive Visualization of Machine Learning Data	204
10.2.1	Guiding Analysts Towards Interactions	205
10.2.2	Analyst and System Understandings of Interactions	208
10.3	Capture Logs	210
10.3.1	The Art of Logging	210
10.3.2	Contextualizing the Interaction	212
10.4	Personalization	213
10.4.1	Predicting Analyst Intent	213
10.4.2	Personalization for Personality	215
10.5	Explainable AI	217
10.5.1	Providing Feedback to the Analyst	217
10.5.2	Interpretability and Uncertainty	219
10.6	User-in-the-Loop Evaluation	221

10.6.1	How to Evaluate	221
10.6.2	Trust	223
10.7	Self-Correction: Overcoming Incorrect Inferences	224
10.8	Discussion	225
10.9	Conclusion	226
11	Discussion and Conclusion	227
11.1	Discussion	227
11.1.1	How Different Are These Operations?	227
11.1.2	“With Respect to What”	228
11.1.3	The HCI and ML Perspectives	229
11.1.4	Design Lessons	230
11.1.5	Generalizing to Other Model Families	232
11.1.6	Future Interactive Algorithm Opportunities	233
11.2	Limitations and Future Work	233
11.2.1	Surveys	234
11.2.2	Tools	234
11.2.3	Study	235
11.3	Conclusion	235
	Bibliography	238

List of Figures

1.1	Implicit clusters formed in the animals dataset [195] using Andromeda [273].	2
1.2	Two annotated layout states from the LightSPIRE study conducted by Endert et al [115]. In the upper layout, the participant used a hybrid clustering approach that combined horizontal temporal organization and vertical topic organization. In the lower layout, the participant clustered documents into independent topic “piles.” Included under Fair Use, 2019.	3
2.1	A PCA projection of the first two principal components of a dataset of 17 dimensions that have been computed from 6,530 Python notebooks [260]. The PCA projection was generated by ClustVis [221]. These two principal components explain 54.4% of the variance in the dataset of 17 dimensions.	9
2.2	An MDS projection of a cereal dataset [165] projected into 2D, as generated by Andromeda [272].	12
2.3	A t-SNE projection (generated by [177]) of the same Python notebook feature dataset as Figure 2.1, this time showing two clear clusters found in the nonlinear manifold that were not apparent in the linear manifold.	13
2.4	Dis-Function [42] uses WMDS to project data into a low-dimensional space. Interactions can be performed directly on the resulting data to provide feedback to the system. © 2012 IEEE.	22
2.5	TopicPanorama [206] presents a topic graph to show topic relationships between documents. © 2014 IEEE.	23

2.6	The feature space transformation technique presented by Mamani et al. [214] permits interactive image classification from user feedback. Included under Fair Use, 2019.	24
2.7	The Clustering Tour interface in Clustrophile [51], permitting analysts to explore possible clustering solutions and a dimension-reduced scatter plot projection of the data. © 2019 IEEE.	26
3.1	Six different user-created clusterings from a States dataset [102] projection. Several of the clusterings (left column) are quite similar, while others (right column) are much more diverse.	29
3.2	Three dendrograms produced using MATLAB from the same input synthetic dataset using single, complete, and average linkage criteria.	32
3.3	The k -means algorithm finds six clusters of animals in the Animals dataset [195], as shown by the Castor system [317].	33
3.4	Although the elbow method shows a total of 9 clusters in this synthetic dataset, plotting the data in three dimensions shows that there are in fact 10 clusters. Clusters #6 and #7 are separable to humans, but separating them computationally does not substantially reduce the total intra-cluster variance.	34
3.5	Rivelo [290] provides analysts with interactive methods for understanding classifier decisions. Included under Fair Use, 2019.	42
3.6	iCluster [101] provides an interface for users to interactively cluster documents, and also suggests additional documents that might fit into existing clusters. Included under Fair Use, 2019.	45

3.7	The iVisClustering system [196] supports a number of cluster operations, including joining, splitting, creating, and removing. Included under Fair Use, 2019.	47
4.1	The Termite system [65] supports the task of identifying clusters spatially and seeing the relative positions of clusters through a matrix view. Included under Fair Use, 2019.	51
4.2	The UTOPIAN system [63] supports cluster labeling for exploration and to support better understanding of the data by showing which terms best describe a given cluster. © 2013 IEEE.	52
4.3	Six different options for pipelines depicting combinations of dimension reduction algorithms and clustering algorithms. In each of these pipelines examples, it is implied that each algorithm could use an independent distance function, resulting in more than just these six pipelines. Further, these pipelines represent a single analysis iteration.	59
4.4	Interactions from the analyst will drive additional executions through the pipeline during the data exploration process. The analyst does not need to select the same pipeline on every iteration of the analysis.	61
4.5	Saket et al. [265] evaluate three options for encoding cluster membership, relating each to the effectiveness of performing node- and group-based tasks. © 2014 IEEE.	64
4.6	Bubble Sets [74] uses the preattentive closure property to display distinct groupings of data, with a clear delineation between what belongs to the cluster and what does not. © 2009 IEEE.	66

4.7	TopicLens [181] takes a dual-encoding approach to visualizing clusters, combining color and position. © 2017 IEEE.	67
4.8	ASK-GraphView [1] uses clustering to enable efficient exploration of very large networks. © 2006 IEEE.	68
4.9	Four options for displaying cluster membership as studied by Jianu et al [168]. In addition to a node-link representation similar to that included by Saket et al., this study included Linesets [9], GMap [129], and BubbleSets [74]. © 2014 IEEE.	70
5.1	A selection of interfaces and tools that support Parametric Interaction or Observation-Level Interaction. The upper row shows PI interfaces that include slider bars from Andromeda (PI view) [273], Star Coordinates [174], and SpinBox widgets from STREAMIT [10]. The lower row shows OLI interfaces from StarSPIRE [38], Paulovich et al. [239], and Mamani et al. [214]. Included under Fair Use, 2019.	81
5.2	The bidirectional, multi-model pipeline for semantic interaction proposed by Dowling et al [97]. Included under Fair Use, 2019.	87
5.3	The “Dimension Reduction Preprocessing for Clustering” projection pipeline from the previous chapter, annotated with the structure of input and output data. In this and many future figures, W=dimension weights, HD=high-dimensional data, LD=low-dimensional data, M=cluster membership.	88
5.4	An analyst repositions the Grizzly Bear observation within the projection, indicated by the orange arrow.	89

5.5	A representation of data flow using a dimension reduction model to learn a new projection. The clustering algorithm is not necessary for observation-to-observation interactions, and could be positioned either to the left or right of the dimension reduction algorithm.	94
5.6	Selection interactions in Andromeda [274]: nearest neighbor selection at the source, radius selection at the target, and additional observation selection in other regions.	94
5.7	A representation of data flow using a Clustering Model to interpret a change in cluster membership, followed by learning distances with a Dimension Reduction Model, to learn a new projection.	97
5.8	A representation of using a Clustering Model alone to interpret a change in cluster membership.	98
5.9	An alternative representation of data flow using a Clustering Model to interpret a change in cluster membership to learn a new projection.	98
5.10	A representation of data flow using a Clustering Model first to interpret a change in cluster membership to learn a new projection.	99
5.11	A representation of data flow in which the Interaction Computations of the Dimension Reduction and Clustering algorithms negotiate an optimal interpretation of the analyst interaction.	100
5.12	A representation of data flow for an interaction that repositions a cluster to another location in the projection, only requiring the interaction computation of the Dimension Reduction Model.	102

5.13	A representation of data flow for an interaction that repositions cluster boundaries to encapsulate new observations, only requiring the interaction computation of the Clustering Model.	103
6.1	The framework and implementation of our cluster-based semantic interaction model.	116
6.2	After reclassifying the Grizzly Bear into the upper-right “Predators” cluster, the Rat, Raccoon, and Weasel all were reassigned out of the “Predators” cluster, and the Polar Bear was removed from the “Pets” cluster.	120
6.3	Four steps through the interaction discussed in the Usage Scenario section. Interactions initiated by the user are shown with blue arrows, while the cluster reassignments initiated by the system are shown with orange arrows.	123
7.1	Clustered projections of three datasets generated by Pollux. From left to right, an Animals dataset [195], the Fisher Iris dataset [102], and a U.S. Census States dataset [305].	128
7.2	The computational pipeline for Pollux. The projection computations convert data into a visualization, while the interaction computations interpret and respond to analyst interactions.	129
7.3	Five different classes of edges that could be included in the layout: Centroid-Centroid Edges (CC), Centroid-Node Internal Edges (CN _I), Node-Node Internal Edges (NN _I), Centroid-Node External Edges (CN _E), and Node-Node External Edges (NN _E).	131
7.4	Mouseover interactions afford a details-on-demand view of the raw data for each observation.	133

7.5	Four views of the Census dataset with a variety of edge class selections. From top to bottom, (A) only CC and CN_I edges, (B) same as above, plus NN_I , (C) same as above, plus CN_E , (D) all edge types.	137
7.6	Nine views of the Census dataset with various CC, CN_I , and NN_I weights. Cluster compactness varies across the x-axis via manipulation of the CN_I and NN_I edge class weights, while pairwise cluster distance varies in the y-axis via manipulation of the CC edge class weight.	138
7.7	(left) A direct two-dimensional projection of the high-dimensional Animals data with cluster information encoded by color. (right) The same data in Pollux, using color encoding for clusters rather than convex hulls.	140
7.8	(left) The Fisher’s Iris dataset with the system-determined two clusters. (right) The analyst updates the view to incorporate three clusters.	141
7.9	Each of the six interactions performed by the analyst in the usage scenario. Nodes enclosed by red rectangles denote analyst-driven classification updates, while nodes enclosed by blue rectangles denote classification updates made by the system in response to newly-learned weights. Lines are drawn to show observation paths from source to destination cluster.	144
7.10	A selection of six attribute weights and their respective value updates during the six analyst interactions.	147
8.1	Gemini pipeline: both the dimension reduction and clustering algorithms are processed on the high-dimensional data in parallel.	152
8.2	After Michigan was assigned to Cluster 2 (green node), both West Virginia and Minnesota were reclassified in response (blue nodes).	154

8.3	The animals dataset [195] loaded into Gemini (left), Castor (center), and Pollux (right). Note that the layout is nearly identical in Gemini and Castor, while the clustering assignments are the same in Gemini and Pollux.	155
8.4	The initial view of the states dataset (top row) and Fisher’s Iris dataset (bottom row) visualized in all three tools.	155
8.5	The projection stress of each dataset while using each tool.	161
8.6	The total intra-clustering distance of each dataset while using each tool. . . .	162
8.7	The Davies-Bouldin Index for each dataset while using each tool.	163
8.8	The normalized change in projection stress and intra-cluster distance over a set of interactions in each tool.	165
9.1	In the Sensemaking Process [248], intelligence analysts transform raw information into reportable results through organizational stages that filter, extract, and structure data. Included under Fair Use, 2019.	172
9.2	A photo of the Killer Whale card in both the (left) labeled and (right) abstract datasets.	175
9.3	One of many potential organizational structures possible to generate from the study dataset.	176
9.4	The radial layout created by Participant A2, in which each of the animals is drawn towards its highest attribute with additional effects by the other large attribute values.	179

9.5	The complex cluster cross-cutting created by Participant A3. This participant judged clusters of abstract data by considering low, medium, and high values of dimensions A, B, and C (corresponding to Furry, Big, and Swims). The axes of the organization were important to the cluster determinations.	180
9.6	Participant L5 created a structure in which the axes were clearly an important feature. Groups were refined based on the remaining three dimensions, but the majority of the structure was governed by the Swims and Big dimensions with which she began her analysis. Attribute bins from the analysis of the Swims dimension are clearly still visible on the x-axis.	181
9.7	Cluster operations in the participant organizational strategies. Cluster join operations destroy two original clusters to create a new cluster, cluster split operations destroy one original cluster to create two new clusters, cluster create operations take a portion of one or more clusters to create a new cluster (while leaving the originals), and cluster remove operations destroy one original cluster and distribute its members to one or more existing clusters.	182
9.8	Time distribution for completing the Organization Task.	184
9.9	The scatter plot created by Participant A5 between the C and E (Swims and Solitary) dimensions.	185
9.10	The spectrum of Furry and Swims created by Participant L2.	186
9.11	Time distribution for completing the Update Task.	187
9.12	Five stages from the analysis produced by Participant L4.	189

9.13	Participant A4 created a complex space, creating spectra for dimensions A and C (Furry and Swims) and distinct regions of high influence for dimensions B, D, and E (Big, Fierce, and Solitary).	193
9.14	Participant A6 created complex hierarchical and cross-cutting cluster structures during her analysis.	194
10.1	Explainable AI focuses on how visualization can be used to support machine learning (a human learns the machine state), while Machine Learning from User Interaction identifies ways by which machine learning can be used to support visualization (a machine learns the human state).	202
10.2	This research agenda captures five interconnected phases in an exemplary human-in-the-loop analytical system framework.	204
10.3	ModelSpace [44], displaying a layout of models and interaction paths. Included under Fair Use, 2019.	206
10.4	The Speculative Execution concept from Sperrle et al [283]. Included under Fair Use, 2019.	207
10.5	Andromeda supports parametric interaction (PI) with interactive slider widgets and observation-level interaction (OLI) with direct manipulations in the projection. Included under Fair Use, 2019.	209
10.6	An example computational pipeline and system from Dowling et al. [97], in which interactions with document relevance are handled by the Relevance Model and interactions with document positioning are handled by the WMDS Model. Included under Fair Use, 2019.	214

10.7	After a system detects frustration, it can display suggested interactions [237]. Included under Fair Use, 2019.	216
10.8	The relationship between suggested and assigned keywords encoded in font size. Included under Fair Use, 2019.	218
10.9	An example computational pipeline and system from Dowling et al. [97], in which interactions with document relevance are handled by the Relevance Model and interactions with document positioning are handled by the WMDS Model. Included under Fair Use, 2019.	220
10.10	The experiment dashboard in HyperTuner [203], allowing an analyst to explore the hyperparameter space. Included under Fair Use, 2019.	222

List of Tables

- 4.1 Sample exploratory data analysis tasks, organized by stage in the data analysis process (rows) and algorithm family (columns). 55
- 4.2 A summary of the design challenges and questions discussed throughout the chapter regarding the combination of dimension reduction and clustering algorithms. 76
- 5.1 Sample interactions, organized by type of interaction (rows) and by the type of algorithm affected by the interaction (columns). 80
- 5.2 A collection of example intents and interactions that an analyst could communicate via repositioning an observation or a cluster in a projection. 91
- 5.3 A summary of the observation interactions discussed in this paper by cardinality, the importance of the interaction source, target, or both, and whether an analyst is thinking high-dimensionally, low-dimensionally, or both. 93
- 5.4 A collection of example intents and interactions that an analyst could communicate via reclassifying an observation with respect to a cluster in a projection that uses cluster boundaries. 96
- 5.5 A summary of the cluster interactions discussed in this paper, again evaluated by cardinality, the importance of the interaction source, target, or both, and whether an analyst is typically thinking high-dimensionally, low-dimensionally, or both. 101

7.1	A summary of the three datasets visualized with Pollux, enumerating each edge type.	135
9.1	The dataset used in the cognitive study described in this chapter.	173
9.2	A summary of the main findings uncovered by this study.	199

List of Abbreviations

AI Artificial Intelligence

AIC Akaike Information Criterion

BIC Bayesian Information Criterion

CC Centroid-Centroid Edges

CFA Confirmatory Factor Analysis

CN_E Centroid-Node External Edges

CN_I Centroid-Node Internal Edges

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DIC Deviance Information Criterion

DPMM Dirichlet Process Mixture Model

DR Dimension(ality) Reduction

EFA Exploratory Factor Analysis

GMM Gaussian Mixture Model

GTM Generative Topographic Map

GUI Graphical User Interface

HCI Human-Computer Interaction

HD High-Dimensional

LAMP Local Affine Multidimensional Projection

LASSO Least Absolute Shrinkage and Selection Operator

LD Low-Dimensional

LDA Latent Dirichlet Allocation

LLE Local Linear Embedding

LMDS Local Multidimensional Scaling

LSA Latent Semantic Analysis

LSI Latent Semantic Indexing

MDS Multidimensional Scaling

ML Machine Learning

MLUI Machine Learning from User Interaction

NN_E Node-Node External Edges

NN_I Node-Node Internal Edges

OLI Observation-Level Interaction

OPTICS Ordering Points To Identify the Clustering Structure

PCA Principal Component Analysis

PI Parametric Interaction

PLP Piecewise Laplacian Projection

PLSI Probabilistic Latent Semantic Indexing

PLSV Probabilistic Latent Semantic Visualization

SIR Sliced Inverse Regression

SOM Self-Organizing Map

t-SNE t-Distributed Stochastic Neighbor Embedding

TF Term Frequency

TF-IDF Term Frequency – Inverse Document Frequency

V2PI Visual to Parametric Interaction

VIS Visualization

WMDS Weighted Multidimensional Scaling

XAI Explainable Artificial Intelligence

Chapter 1

Introduction

In recent years, analysts have worked to explore and draw conclusions from increasingly larger datasets, growing both in cardinality and dimensionality. Visual metaphors for exploring high-dimensional datasets come in a variety of forms, each with their own strengths and weaknesses in both visualization and interaction [121, 227]. In particular, datasets with high dimensionality present tractability challenges for computation, design, and interaction [92]. One frequently-used method of visual abstraction is to reduce a high-dimensional dataset into a low-dimensional space while preserving properties of the high-dimensional structure (e.g., retain or respect pairwise relationships from the higher dimensions in the lower dimensional projection). Such dimension reduction algorithms are useful abstractions because some of the dimensions in the dataset may not be essential to understanding the underlying patterns in the dataset [124]. Instead, a subset of the dimensions can be selected or learned (or new dimensions introduced) to define the important characteristics of the dataset. The visualization tasks associated with dimension reduction algorithms have been well studied [40, 41].

Because many dimension reduction algorithms rely on a “proximity \approx similarity” metaphor, attempting to preserve high-dimensional distances in the low-dimensional space, groups of similar observations¹ become positioned close together in low-dimensional space (and ob-

¹In this dissertation, I employ the convention of referring to the features (columns) of a dataset as *dimensions*, individual data items (rows) as *observations*, and the features of those observations (cells) as *attributes*. *Node* is used to indicate the glyph representing an observation in a visualization.

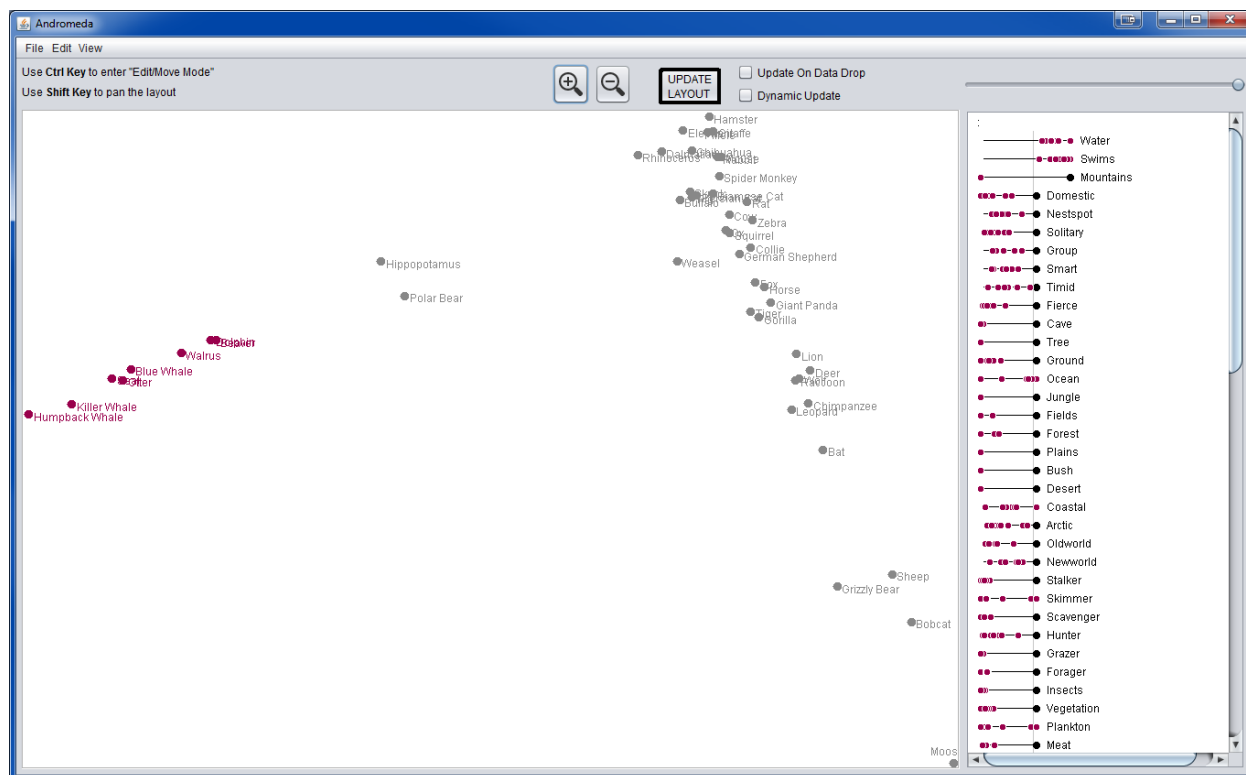


Figure 1.1: Implicit clusters formed in the animals dataset [195] using Andromeda [273].

servations that are dissimilar are spaced further apart). Thus, implicit clusters of similar items form in the projection. This stands in contrast to explicit clusters as identified by clustering algorithms. An example of these implicit clusters is shown in Figure 1.1 using Andromeda [273] with the animals dataset [195]. By increasing the weight on the Water, Swims, and Mountain dimensions equally, two major groups of animals emerge, along with a few outliers.

Studies have linked dimension reduction algorithms to clustered data; for example, Choo et al. discusses dimension reduction methods for two-dimensional visualization of high-dimensional clustered data, proposing a two-stage framework for visualizing such data based on dimension reduction methods [62]. Indeed, clustering can even be thought of as extremely low-resolution dimension reduction, where knowledge about the various attributes

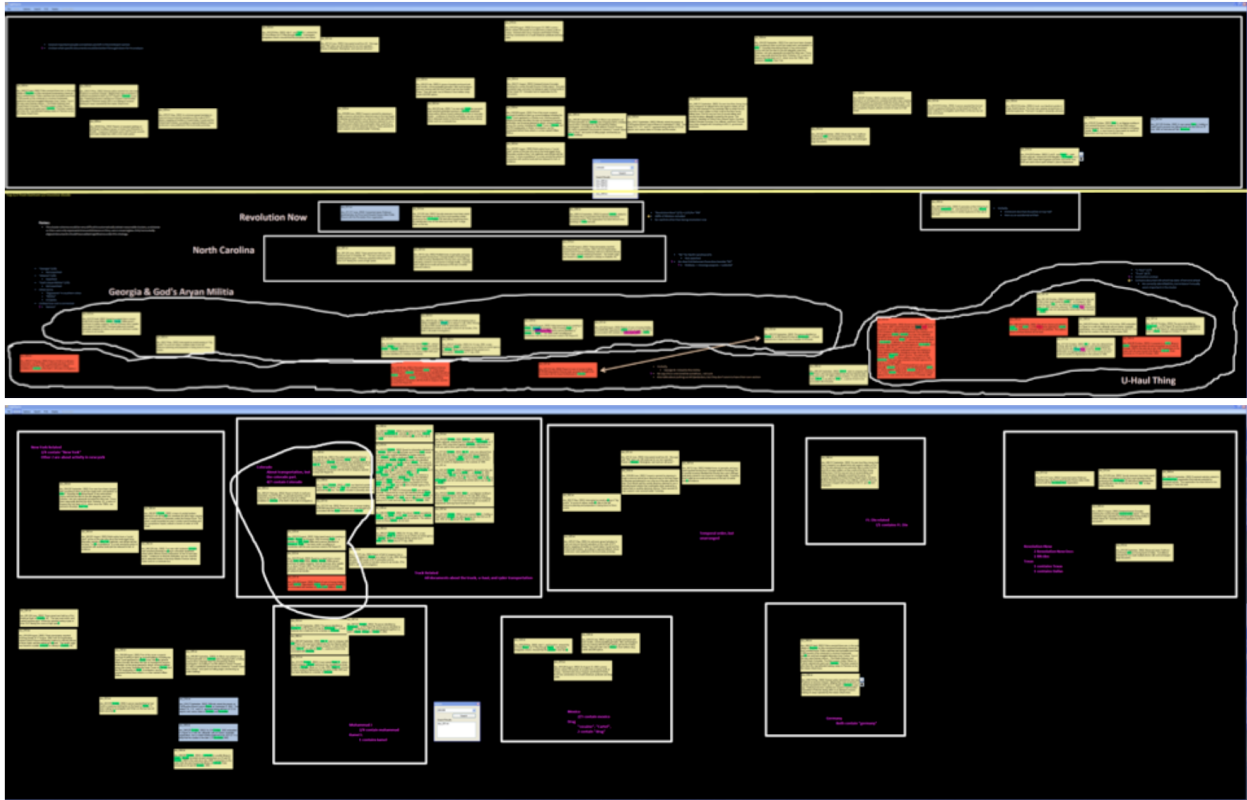


Figure 1.2: Two annotated layout states from the LightSPIRE study conducted by Endert et al [115]. In the upper layout, the participant used a hybrid clustering approach that combined horizontal temporal organization and vertical topic organization. In the lower layout, the participant clustered documents into independent topic “piles.” Included under Fair Use, 2019.

of the observations leads to a one-dimensional bin assignment (or a set of probabilities for multiple bin assignments). This relationship between dimension reduction and clustering is also supported mathematically in specific instances: Ding and He [89] proved that principal components are the continuous solutions to the discrete cluster membership indicators for k -means clustering, indicating that Principal Component Analysis dimension reduction implicitly performs clustering as well.

However, clusters are inherently subjective structures, making their identification by both humans and machines a challenging process that is often problem-specific. Previous research

has shown that humans use a variety of organizational principles to cluster information [96], even when addressing the same task [15]. For example, Endert et al. found that while many users adopt a topic clustering approach to create piles of similar documents, some take other approaches such as creating timelines [115] (see Figure 1.2). A clustering algorithm such as k -means, which naturally detects convex clusters with similar covariance, will likely handle the topics case but not necessarily the timeline case. In order to computationally identify clusters, hundreds of clustering algorithms have been implemented, each with strengths and weaknesses. Indeed, there is no universally-optimal clustering algorithm. Instead, the best clustering algorithm to solve a problem is often determined experimentally [118]. Therefore, introducing a clustering algorithm into a dimension-reduced projection remains a challenge for visualization developers and designers.

From this background information, I establish the following overarching research challenge: visualization creators developing interactive projection tools must select dimension reduction and clustering algorithms that suit the exploration goals of the analyst and the properties of the underlying data, while also operating cooperatively with each other and providing meaningful insight back to the analyst. This is certainly a broad area of research, well beyond the scope of a single dissertation. While providing a general survey of dimension reduction and clustering, I primarily focus on presenting the dimensionally-reduced data in the form of a two-dimensional interactive projection. Clusters can then be embedded within these projections. Further, I focus on surveying and producing tools that contain a single view of the data at one time, which can be updated based on what the system has learned about the interests of the analyst who interaction with the system. In this work, I investigate this systematic combination of dimension reduction and clustering algorithms. This research covers the following points of investigation:

- What are the commonly-used dimension reduction algorithms, what are their properties, and how are they used in interactive applications? (Chapter 2)
- What are the commonly-used clustering algorithms, what are their properties, and how are they used in interactive applications? (Chapter 3)
- When combining dimension reduction and clustering algorithms into the same interactive visual analytics tool:
 - What possibilities exist for creating projections? (Chapter 4)
 - What possibilities exist for interaction with those projections? (Chapter 5)
- Selecting some sample points from this design space:
 - How can a dimension reduction-first application that supports semi-supervised learning be designed? (Chapter 6)
 - How can a clustering-first application that supports semi-supervised learning be designed? (Chapter 7)
 - What are the benefits to each of these approaches, both from a computational standpoint and from an analyst hoping to gain insight? (Chapter 8)
- How do analysts cognitively group and spatialize, and how can interactive tools best support these cognitive processes? (Chapter 9)
- What research challenges remain to be solved in the area of intelligent human-in-the-loop systems? (Chapter 10)

Some of the contributions presented in the following chapters include:

- A presentation of the variety of dimension reduction and clustering algorithms that have not yet been integrated into interactive visual analytics tools (Sections 2.1 and 3.1).
- A collection of projection pipelines for transforming input data into a visualization via dimension reduction and clustering models. (Figure 4.3).
- Three tools that implement these projection pipelines to support human-in-the-loop

interactive analytics: Castor (Chapter 6), Pollux (Chapter 7), and Gemini (Chapter 8).

- Design decisions to be addressed when creating a dimension reduction and clustering visualization (Table 4.2 and Section 8.4.2), factors to consider when interpreting the intent of an interaction (Section 5.5.4), and methods for extending the design of existing tools (Section 7.2).
- The design and implementation of a study to investigate the cognitive behaviors performed by analysts when organizing, structuring, and grouping data (Chapter 9).
- A research agenda to develop future intelligent human-in-the-loop analytical tools and techniques (Chapter 10).

Chapter 2

Background: Dimension Reduction

Algorithms and Tools

This background chapter presents a brief survey of dimension reduction algorithms (Section 2.1) and tools (Section 2.2). The goal of dimension reduction algorithms is to represent high-dimensional data in a low-dimensional space while preserving high-dimensional structures, including outliers and clusters [197]. Both humans and machine learning algorithms struggle to comprehend and process high-dimensional data (the “Curse of Dimensionality” [27]), and analysis by both parties can be made more efficient by reducing data into fewer dimensions [156, 301, 343]. Dimension reduction has a scalability advantage over other methods for visualizing high-dimensional data such as parallel coordinate plots and heatmaps, but with the disadvantage of information loss when transforming the data into the low-dimensional projection [121, 207, 227]. Here, we summarize many of the common dimension reduction algorithms and approaches in the visualization field; more detailed surveys of dimension reduction algorithms and tools can be found in the literature [124, 126, 197, 264, 327]. In addition, several tools have been implemented that allow analysts to switch between and compare dimension reduction algorithms [205, 252, 281].

2.1 Dimension Reduction Algorithms

A common method for discussing dimension reduction algorithms is to divide them into linear and nonlinear classes, referring to the structure of the underlying manifolds or topological spaces that each class can learn. Linear dimension reduction algorithms are limited to learning linear (or affine) manifolds, while nonlinear dimension reduction algorithms can learn more complex manifolds. Still other dimension reduction algorithms have been implemented in both linear and nonlinear variants. We begin this chapter by detailing several commonly-used dimension reduction algorithms in the visualization literature. Following this brief survey, we discuss several specific categories of dimension reduction techniques, including topic modeling, feature selection, and feature learning. The section concludes with a discussion of distance functions.

2.1.1 Linear Dimension Reduction

Principal Component Analysis (PCA) is a commonly-used technique for linear dimension reduction. PCA works by determining the axes of maximum variance in the collection of observations [171, 240]. Mathematically, this consists of an orthogonal linear transformation that recasts the input data into a new coordinate system in such a way that the axis of maximum variance lies on the first principal component, the axis of second greatest variance lies on the second principal component, and so on. This transformation is defined by a set of orthonormal p -dimensional weight vectors \mathbf{w} that map each row of the input data \mathbf{x} to a new principal component vector \mathbf{t} , given by $\mathbf{t} = \mathbf{x} \cdot \mathbf{w}$ so that the individual values $t_{1..m}$ inherit the maximum possible variance from \mathbf{x} . After scaling and centering the data, the maximizing vectors are given by the eigenvectors of the data's covariance matrix (or, equivalently, the right singular vectors of the data stacked as rows in a matrix).

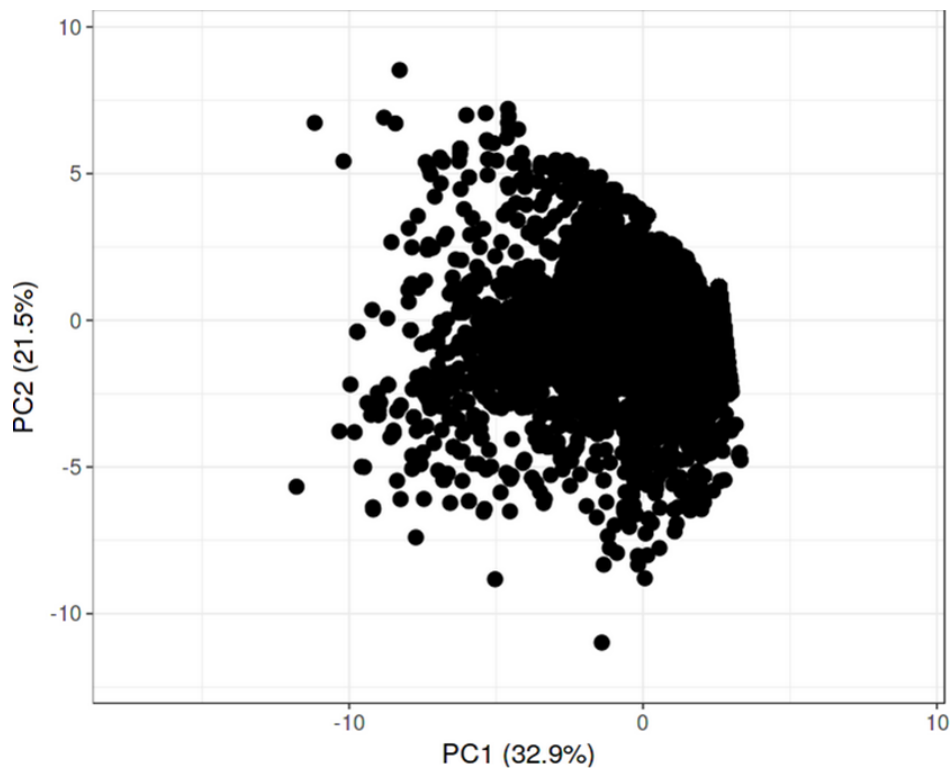


Figure 2.1: A PCA projection of the first two principal components of a dataset of 17 dimensions that have been computed from 6,530 Python notebooks [260]. The PCA projection was generated by ClustVis [221]. These two principal components explain 54.4% of the variance in the dataset of 17 dimensions.

There are as many principal components as there are dimensions in the original dataset; therefore, for dimension reduction, only the first L principal components must be computed. Selecting $L = 2$ and keeping only the first two principal components thus produces a two-dimensional projection of the high-dimensional data with maximal variance along the first two principal components, as shown in Figure 2.1. An important clustering note is that clusters may be spread out in the projection as a result of the data spread, making them more evident in the projection. In contrast, selecting two random vectors for projection could lead to overlap in these clusters, making a pair of clusters indistinguishable.

While traditional PCA is a linear technique, the approach can be generalized into nonlinear dimension reduction techniques. Principal curves and manifolds [133] extend PCA, as does

multilinear PCA in multilinear subspace learning [210]. A probabilistic variant of PCA [299] has also been implemented. Further, there exist non-linear implementations of PCA, but the non-linearity exists in the objective function; the resulting principal components remain as linear combinations of the original variables [176].

Factor analysis [142] is a linear dimension reduction technique that is similar but not identical to PCA. While both techniques examine the variability of a dataset, factor analysis assumes that an underlying structural or causal model exists, while PCA is simply a variable reduction technique. There are two primary types of factor analysis. Exploratory factor analysis (EFA) is used to identify relationships within a dataset and to group similar items without any *a priori* guess as to what these might be, while confirmatory factor analysis (CFA) includes hypothesis testing to validate that observations are associated with specific factors [295]. As a result, PCA is often used as a method for EFA feature extraction, identifying a sequence of axes with maximal variance.

Another linear technique is projection pursuit [127, 192]. The goal of this technique is to find the most interesting projections in a high-dimensional dataset, where “interesting” is often defined as those projections least similar to a Gaussian distribution. By locating such interesting low-dimensional projections, structures such as clusters and surfaces can be extracted and analyzed while noisy and information-poor variables are ignored. This idea has been extended to projection pursuit regression [128] for data analysis and targeted projection pursuit [120] for feature selection and visualization.

2.1.2 Nonlinear Dimension Reduction

Because high-dimensional data is comprised of many separate and potentially interrelated factors, it is often difficult to interpret. The complexity of the data often signifies that a

linear manifold is insufficient to accurately capture the data of interest. As such, a number of techniques have been developed to extend linear methods in order to identify non-linear spaces that can still be represented in a low-dimensional projection.

The most commonly-used technique for nonlinear dimension reduction has historically been Multidimensional Scaling (MDS) [193, 300], though other nonlinear techniques have become more influential in recent years. The goal of MDS is to visualize similarities between observations within a dataset. This typically involves transforming a high-dimensional dissimilarity matrix into a low-dimensional space that minimizes a strain loss function, thereby minimizing spatial distortion. As a result, these measures of similarity between high-dimensional observations are generally preserved in the low-dimensional projection, further causing high-dimensional structures and features such as clusters and outliers to persist in the low-dimensional space [35]. This low-dimensional space is typically selected as 2D to visualize the dimension-reduced data in a coordinate plane (see Figure 2.2), though 3D can also be used for immersive spaces [311].

To be precise, MDS begins by calculating the distances between each pair of points. Store these in a matrix \mathbf{D} such that $\mathbf{D}_{i,j}^x = d(\mathbf{x}_i, \mathbf{x}_j)$ where $d : \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty]$ is a function giving some notion of (dis)similarity between points (perhaps most common among these would be the Euclidean or ℓ_2 distance, given by $d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{p=1}^P (x_{i,p} - x_{j,p})^2}$). It then tries to find a set of low dimensional data $\mathbf{z}_i \in \mathbb{R}^L$ which induce a pairwise low-dimensional dissimilarity matrix D^z which best matches D^x . In particular, the following loss functional is minimized:

$$\mathbf{Z} = \underset{\mathbf{Z}}{\operatorname{argmin}} l(\mathbf{Z}) = \underset{\mathbf{Z}}{\operatorname{argmin}} \|\mathbf{D}^z - \mathbf{D}^x\|_F \quad (2.1)$$

where $\|\mathbf{A}\|_F$ gives the Frobenius norm of \mathbf{A} (i.e. the root of the sum of its square elements, or simply its norm when treated as a vector). This minimization is rendered nontrivial by

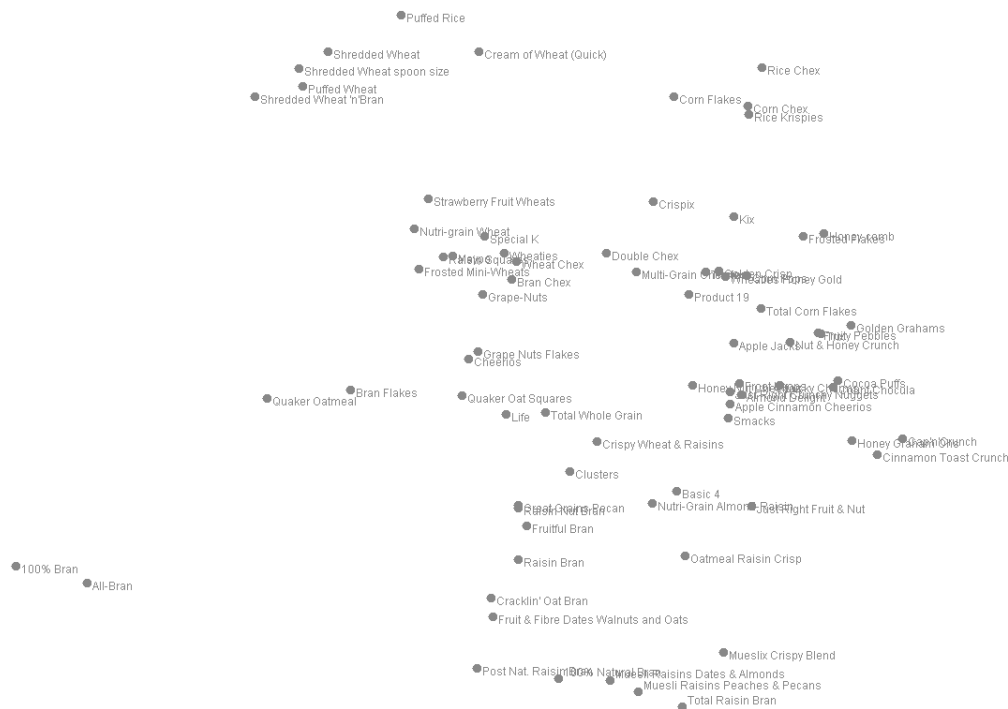


Figure 2.2: An MDS projection of a cereal dataset [165] projected into 2D, as generated by Andromeda [272].

the non-convexity of l : in general, only local optima should be expected. This objective may be minimized via standard gradient-based optimization techniques; iterative quadratic majorization has been popular historically.

The MDS technique has been extended in a variety of methods. Weighted Multidimensional Scaling (WMDS) [50, 270] introduces a weight vector that is applied to the dimensions, influencing the high-dimensional distances and hence the low-dimensional projection that results from the application of the algorithm. Isomap [294] assumes that high-dimensional distances are only known between neighboring observations rather than between all pairs, using this neighborhood graph to compute an estimate of the full dissimilarity matrix before calling classic MDS to compute the low-dimensional coordinates for the full dataset. Glimmer [164] is an MDS implementation that is designed to exploit parallelism on GPUs, improving the computational efficiency while also organizing the input data into a hierarchical structure



Figure 2.3: A t-SNE projection (generated by [177]) of the same Python notebook feature dataset as Figure 2.1, this time showing two clear clusters found in the nonlinear manifold that were not apparent in the linear manifold.

that makes solutions at local minima less likely. CLARET [80] improves upon Glimmer and WMDS with additional preprocessing of the observations, extending the overall MDS technique to much larger data scales. Local Multidimensional Scaling (LMDS) [309] creates localized regions of the high-dimensional spaces that are dimension-reduced using MDS, with the full space then combined via convex optimization.

Supplanting MDS in recent visualization research is t-distributed Stochastic Neighbor Embedding (t-SNE) [211] (see Figure 2.3). t-SNE functions by modeling each high-dimensional

observation as a multidimensional probability distribution so that similar observations are modeled by nearby observations and dissimilar observations are modeled by distant observations. A similar probability distribution is constructed in the low-dimensional space, and the relative entropy between the two distributions is minimized. Similar to LMDS, t-SNE performs different transformations on different regions of the data, controlled by a *perplexity* parameter to balance local and global high-dimensional features. Selecting the optimal perplexity value is a substantial challenge for using t-SNE [316], leading to the development of visualization tools to inspect and explore t-SNE projections [54], as well as for methods to automatically learn optimal perplexity values [48].

Self-organizing maps (SOMs) [184, 185] are neural networks that are trained to produce low-dimensional representations of high-dimensional samples, making use of a neighborhood function similar to that of Isomap to preserve the structure of the high-dimensional space. The SOM learning algorithm begins with the neuron weights initialized to random or evenly-sampled values. These weights are then updated as the overall model is trained using a competitive learning process [261], in which the neurons compete to respond to portions of the input data. In the SOM case, the neuron with the weight vector most similar to a training input vector is adjusted towards the input vector in an iterative process across a sequence of training input. The neighbors of this best matching neuron are also adjusted towards the training vector, thereby influencing larger regions of the neural network. The process iterates until a predetermined limit is reached. SOMs also are useful as a dimension reduction technique which can be interpreted as a set of clusters without the need for intermediate feature transformation.

In particular, SOM consists of a set of R neurons, each with a weight vector \mathbf{w}_r of length p and together with a topology giving some sense of closeness between the neurons. For the purpose of two dimensional visualization this is typically a 2D lattice with adjacent neurons

declared as neighbors. Training proceeds iteratively by choosing one data vector at a time and computing its dot product with each neuron’s weight vector. The neuron which has the greatest inner product with that data vector is then made to look more like it, as are its neighbors. Various update rules are available, the original being:

$$\mathbf{w}_r = \frac{\mathbf{w}_r + \alpha \mathbf{x}_i}{\|\mathbf{w}_r + \alpha \mathbf{x}_i\|} \quad (2.2)$$

where \mathbf{w}_r is the “winning” neuron or one of its 8 nearest neighbors on the lattice, \mathbf{x}_i the current datapoint being examined, and α is a hyperparameter giving the learning rate. In the originator’s implementation, weights are scaled to have Euclidean norm 1.

2.1.3 Topic Modeling

The goal of topic modeling is to generate a set of abstract topics that occur in a collection of documents. Such techniques reduce dimensionality by converting a significant number of words to a small number of topics (or more generally, defining an unobserved set of groups to interpret a set of observations), while also clustering documents with similar words under overarching topics. Because documents can contain multiple topics, each document has a probability of being assigned to a particular topic, making this a soft clustering assignment [314].

Latent Semantic Indexing (or LSI, also referred to as Latent Semantic Analysis or LSA) [238] was an early topic model. It involves calculating the singular vectors of some matrix encoding a corpus, for instance the Term Frequency (TF) or Term Frequency-Inverse Document Frequency (TF-IDF) matrices, which are then treated as topics, and is closely related to PCA. As singular vectors are real valued, there is some difficulty in interpreting topics (what are we to do with the fact that Topic 4 has a value of -0.635 associated with the

word “airplane”?). Hofmann [152] introduced a technique called Probabilistic LSI (PLSI) which improved model interpretability by constraining the latent topics to be nonnegative. Each topic is represented as a probability distribution over words (now we say that Topic 4 produced documents which contain the word “airplane” some proportion of the time). The PLSI model has several parameters for each distinct word in the vocabulary, and as such, can suffer from overfitting issues, especially with rarer words. A Bayesian solution to this issue was proposed by Blei [34] with the introduction of their Latent Dirichlet Allocation (LDA) model, the result of placing prior distributions over the PLSI model. In LDA, documents are treated as a mixture of a small set of topics assumed to have a Dirichlet prior distribution, each incorporating a small set of frequently-used words. Most commonly, a variational Bayes inference process is then used to infer the full collection of distributions: the set of topics, the associated word probabilities ϕ , the topic of each word \mathbf{Z} , and the topic mixture of each document θ . Other techniques are available for statistical inference on topic models, most prominently Collapsed Gibbs Sampling; see Asuncion [19] for a comparative study.

2.1.4 Feature Selection

Feature selection refers to the process of selecting a subset of the input dimensions for use in future processing, relying upon the assumption that the dataset contains some dimensions that are irrelevant or redundant. As a result, these dimensions can be removed without significant impact [29]. Feature selection differs from techniques discussed previously in this chapter in that the result is a subset of the input features, while other techniques transform the input into new features. Feature selection techniques are often divided into three categories: filter, wrapper, and embedded methods [138].

The goal of filter methods is information gain, attempting to remove the effects of the least

interesting dimensions and use those that remain for further modeling. Identifying uninteresting dimensions can be as straightforward as finding similar features through correlation measures [245], and such techniques are computationally efficient due to their simplicity.

Wrapper methods build and train a series of models using a subset of the original dimensions, then test these models on ground truth data to score the quality of each subset. This is clearly a computationally-intensive approach and can be prone to overfitting with small numbers of observations, but enables the ability to detect interactions between variables [245]. Wrapper methods can occasionally use filter methods for a preprocessing stage, as seen in the Recursive Feature Elimination algorithm [139], which can allow wrapper techniques to be used at larger scales.

Finally, embedded methods perform feature selection as a component of the model building process. For example, the LASSO method [297] in regression analysis includes a penalty on model coefficients, effectively selecting those features that have non-zero coefficients. Features are selected for removal while constructing the model, approaching the computational efficiency of filter methods while incorporating the model-based tests of wrapper methods.

2.1.5 Feature Learning

Feature learning (or representation learning) techniques incorporate the ability for a system to discover the appropriate representations needed to complete a task [28]. While feature learning techniques are not specific to dimension reduction, a number of unsupervised techniques fall under this category. For example, PCA is a linear feature learning technique, creating axes of large variance from the input data. Similarly, Local Linear Embedding (LLE) [259] is used to generate low-dimensional representations from high-dimensional data by searching for high-dimensional vectors that minimize the distance between and observa-

tion and its low-dimensional representation.

Autoencoders are a neural network-based feature learning technique that are trained to ignore noise and focus on signal, and hence can be used create a useful low-dimensional representation that ignores irrelevant input [204]. Indeed, one of the early motivations to study autoencoders was an interest in applying deep learning techniques to dimension reduction problems, with even the earliest attempts outperforming PCA in both reconstruction error and interpretability [132, 151].

Generative topographic maps (GTMs) are similar to SOMs, though using a probabilistic model with a Gaussian noise assumption [32]. Like the SOM approach, this is an iterative process in which a low-dimensional point is mapped to the high-dimensional space, with noise subsequently added into that space. As a result the introduction of Gaussian noise, a Gaussian mixture model is created, which can then be learned by expectation maximization [226].

2.1.6 Sufficient Dimension Reduction

In statistics, a popular way to handle dimensionality reduction in the supervised case is motivated by the notion of *conditional independence*. Two random variables X and Y are said to be conditionally independent given a third Z if, intuitively, Z contains all information about X relating to Y , or, (somewhat more) rigorously, $P(X \cap Y|Z) = P(X|Z)P(Y|Z)$ (a fully rigorous definition is surprisingly difficult, see Billingsley [31]). Sufficient dimension reduction thus attempts to find a low dimensional representation Z of the random variable X which contains all information relating X and Y in the sense of conditional independence. The most popular methods for supervised dimension reduction involve choosing a linear combination of the input variables, that is, $Z = \mathbf{A}X$ for a suitable matrix \mathbf{A} , naturally

resulting in linear dimension reduction. Methods differ primarily in their technique for estimating \mathbf{A} .

Sliced Inverse Regression, or SIR [201], was the first principle method in this area. Suppose we have observed (potentially high dimensional) features $\mathbf{X} \in \mathbb{R}^{N \times P}$ and a response vector $\mathbf{y} \in \mathbb{R}^N$. SIR proceeds by sorting the responses y_i into B many bins \mathcal{B}_i . Within each bin, take the average of the corresponding rows of \mathbf{X} to form $\bar{\mathbf{x}}_i = \frac{1}{|\mathcal{B}_i|} \sum_{j \in \mathcal{B}_i} \mathbf{x}_j$, stacking them as rows in a matrix $\bar{\mathbf{X}} \in \mathbb{R}^{B \times P}$. As in PCA, the right singular vectors are chosen as \mathbf{A} . Analysis of the singular values of $\bar{\mathbf{X}}$ can help the analyst determine the correct size for the reduced space.

The same author also proposed the Principle Hessian Directions method [202], which concerns itself with the eigendecomposition of the Hessian matrix of the mapping between \mathbf{X} and \mathbf{y} . This may be estimated without any first or second derivative information via Stein's Lemma, the computation for which involves a generalized eigenvalue problem.

The Active Subspace Method [75] is a related but distinct supervised dimension reduction technique which has recently exploded in popularity in the applied math community. It applies to the case where gradient information is available and the relationship may be sampled at any desired \mathbf{x}_i point. It simply involves stacking gradients as rows in a matrix \mathbf{G} then examining that matrix's right singular vectors. Similar ideas have been explored in the statistics community since at least Samarov in 1993 [266].

2.1.7 Distance Functions

Many of the dimension reduction algorithms surveyed in this section require a distance function as input, which provides the method for calculating the similarity or dissimilarity of each pair of observations. Much like the breadth of algorithms discussed, a number of

distance functions are used in visualization systems. The most popular metrics are those derived from p -norms, which give distance functions of the form

$$d_p(\mathbf{x}_i, \mathbf{x}_j) = \left(\sum_k |x_{i,k} - x_{j,k}|^p \right)^{1/p}.$$

Such a distance is defined for any positive p . The most familiar examples are $p = 1$, which is known as Manhattan distance (due to the city's regular grid structure), and $p = 2$, Euclidean distance. Aggarwal et al. [5] showed that Manhattan distances are preferable to Euclidean distances for high-dimensional data, as Euclidean distance (and p -norms of $p > 1$ in general) tends to compress the space as more dimensions are added, resulting in high-dimensional distances that are less distinguishable. In determining the appropriate distance function, it is also worth considering that some distance functions are computationally more difficult when optimizing a stress function.

Beyond these standard distance measures are others that handle special cases and are less frequently seen. For example, Cosine Distance is often used for measuring distance between documents that have been converted to TF-IDF feature vectors, as these feature vectors are often sparse with a large number zero-value attributes [280]. Similarly, Gower Distance is useful for computing distances for which some attribute measures are missing, as well as for computing distances in datasets with mixed variable types [134].

Many of the distance functions that are applied to dimension reduction algorithms are also useful when considering cluster computations. Cosine distance, for example, can be used to measure cohesion within clusters [291]. The Jaccard similarity coefficient is used for measuring diversity and dissimilarity between clusters or sets of observations [200]. Mahalanobis distance, which measures a distance between a point and a distribution, also frequently finds use in cluster and classification analysis [219].

2.2 Interactive Dimension Reduction Tools

Systems that provides interfaces to interact with dimension reduction algorithms are nearly as diverse as the algorithms themselves. The semantic interaction work initiated by Endert et al. [110, 111, 112, 114, 116] has led to much of the current research into interactive dimension reduction tools. Under this semantic interaction paradigm, incremental feedback is delivered to the system by an analyst [277]. The system then uses these interactions to infer the intent of the analyst and to update the projection accordingly, often via adjustments to a vector of weights applied to the dimensions of the dataset. Research from Brown et al. has demonstrated that such interaction sequences can be used to learn characteristics about the behavior of analysts [43]. A brief survey of systems that follow similar approaches is presented by Boukhelifa et al [37].

These incremental dimension reduction tools can be divided into classes that support quantitative data, text data, and more complex data. The quantitative data case is the most straightforward, as dimension reduction algorithms process numerical data and distances by default. The text and complex data cases are special variants of the quantitative case, as this data must be processed into numerical data before running the dimension reduction algorithms. We start the discussion of interactive tools in this section by briefly surveying these three categories. From there, we discuss tools that support big datasets, tools designed for immersive and 3D spaces, and conclude with a brief discussion of the relationship between dimension reduction and clustering.

2.2.1 Quantitative Data

Leman et al. [199] introduced a new human-data interaction paradigm called “Visual to Parametric Interaction” (V2PI), demonstrating a method by which visualizations of high-

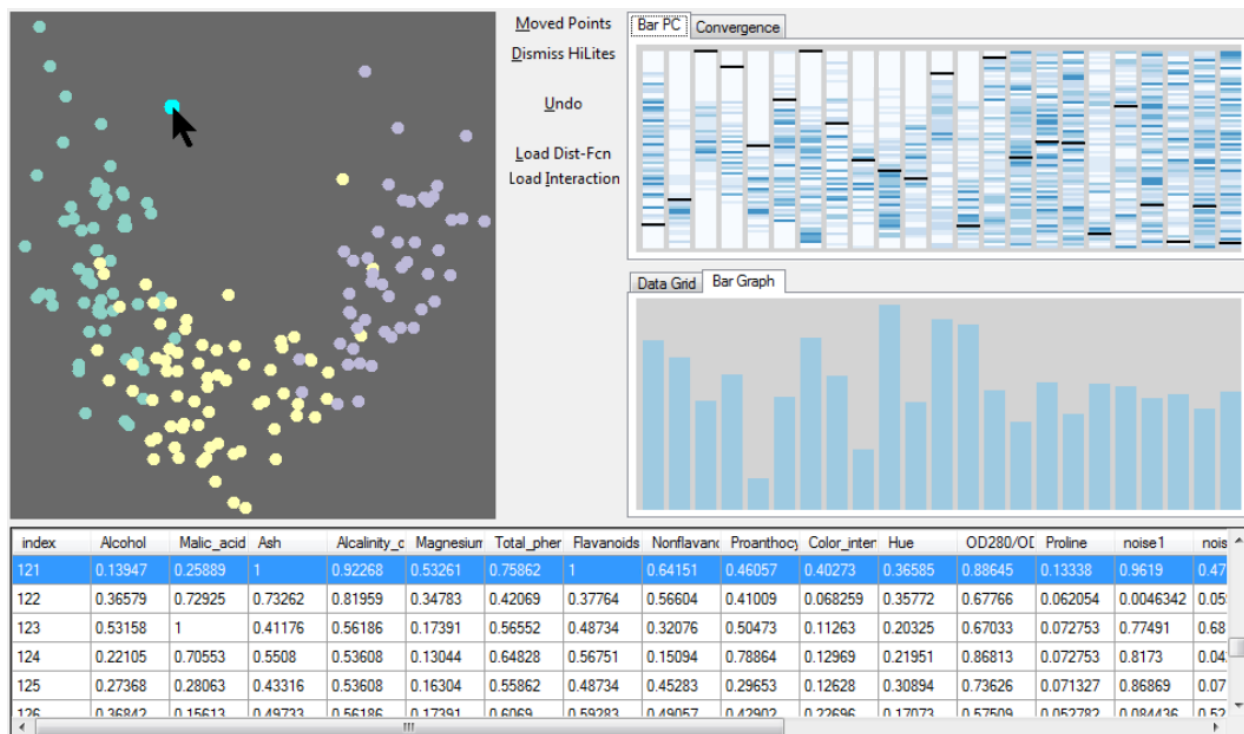


Figure 2.4: Dis-Function [42] uses WMDS to project data into a low-dimensional space. Interactions can be performed directly on the resulting data to provide feedback to the system. © 2012 IEEE.

dimensional data can adjust to expert feedback. Most importantly, this paradigm allows for analysts to provide feedback by directly manipulating the visualization rather than configuring models and parameters; instead, the system uses the expert feedback to generate a new set of model parameters. This interaction approach serves as the basis for a number of interactive dimension reduction systems. Andromeda [273], introduced in the previous chapter (see Figure 1.1), uses WMDS as a dimension reduction method to render high-dimensional quantitative data into an interactive projection, which can then be manipulated by the user to learn a new set of parameter weights. A similar approach is taken by Dis-Function [42] (see Figure 2.4), which also supplements the low-dimensional projection with additional views.

In contrast, Molchanov et al. [225] introduce a PCA-based technique, using user-manipulated control points as feedback to solve a system of linear equations that generates a new pro-

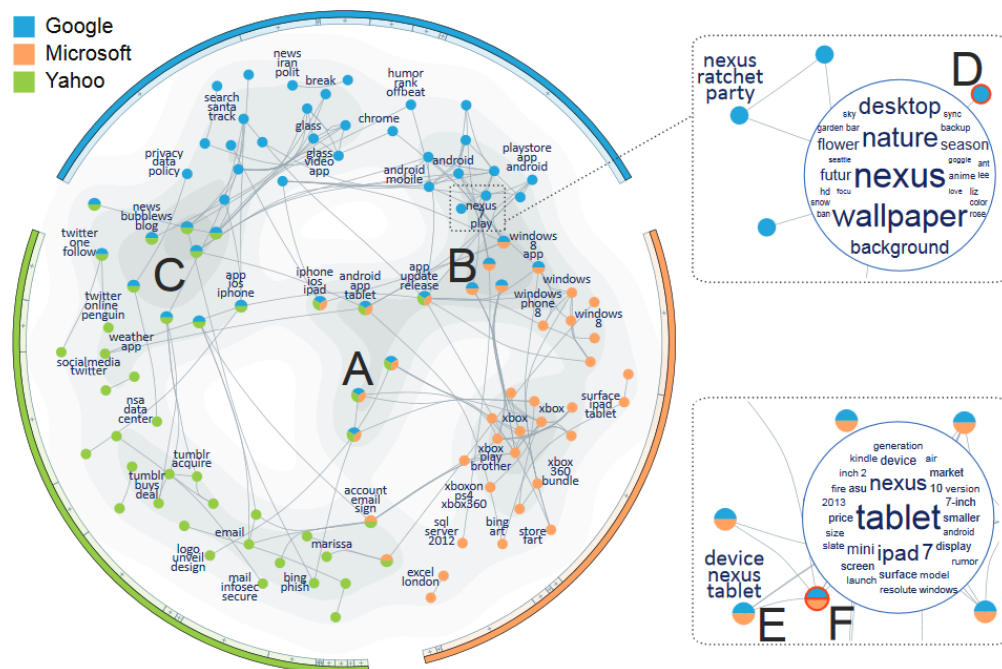


Figure 2.5: TopicPanorama [206] presents a topic graph to show topic relationships between documents. © 2014 IEEE.

jection. A similar PCA-based technique is implemented in EvoGraphDice [47], permitting an analyst to progressively modify the projection via an interactive evolution process. SIR-IUS [98] provides interactive projections of both the observations and the dimensions of a dataset, allowing for correlations to be visualized and manipulated in both panels. Further, interactions do not need to be focused on the observations in the projection. Instead, analysts can interact with the axes [180] and the projection space [194].

2.2.2 Text Data

Inspired by document projections generated with the IN-SPIRE system [328], a number of interactive tools have been implemented to structure text documents using a similar direct manipulation feedback process. This necessitates the conversion of the text into quantitative values for the dimension reduction algorithms to manipulate, often taking the form of term



Figure 2.6: The feature space transformation technique presented by Mamani et al. [214] permits interactive image classification from user feedback. Included under Fair Use, 2019.

frequencies (TF) or term frequencies scaled by the frequency of that term appearing in the overall corpus (TF-IDF). Accompanying the introduction of semantic interaction is ForceSPIRE [114], which uses term co-occurrence and a force-directed layout to visually structure a document collection. StarSPIRE [38, 319] extends ForceSPIRE with the introduction of a foraging mechanism, now requiring updates to multiple computational models from the analyst feedback.

TopicPanorama [206] (Figure 2.5) introduces a graph-based representation to visualize topic assignments of documents. Expert feedback permits users to interactively modify and analyze the graph. The CorpusViewer interface in the Serendip system [8] uses an interactive matrix representation, allowing analysts to search for patterns in topic-document relationships through ordering and selection interactions.

2.2.3 Complex Data

Interactive dimension reduction techniques have been extended to more complex data, which again must be converted into quantitative features for the algorithms. Interactive projections of images have been produced with the feature space transformation technique introduced

by Mamani et al. [214] (Figure 2.6), the Piecewise Laplacian Projection (PLP) technique from Paulovich et al. [239], the Local Affine Multidimensional Projection (LAMP) system from Joia et al. [169] (made interactive by the iLAMP extension [93]), and the photo feature space used in the SelPh system [186]. In each of these systems, interactive feedback from an expert can be used to demonstrate learned similarity between images. LAMP and PLP have also been demonstrated on audio files. The Drag and Track interface demonstrated by Orban et al. [234] can be used to explore the parameter space of computational simulations.

2.2.4 Big Data

Large datasets present performance difficulties with some dimension reduction algorithms. For example, MDS requires a distance to be computed between each pair of observations, resulting in $\sim n^2/2$ distances computed for n observations. However, tools do exist to visualize such large datasets. For example, ASK-GraphView [1] supports the visualization of graphs with up to 200,000 nodes and 16,000,000 edges by using clustering to construct a hierarchical graph, thus visualizing only internal subsections of the graph at any time. Visual analytics tools can also be extended to larger datasets through connections to external search services. StarSPIRE has been extended with web integration to the Bing search engine and to IEEE Xplore, modifying its existing foraging routine to access external data [39]. Similarly, Cosmos [99] uses the Elasticsearch library to provide access to remote information repositories of documents, foraging and display relevant search results using WMDS.

2.2.5 Immersive and 3D Spaces

While many interactive dimension reduction systems reduce the low-dimensional space to 2D, immersive analytics provides a new research environment with an extension into 3D

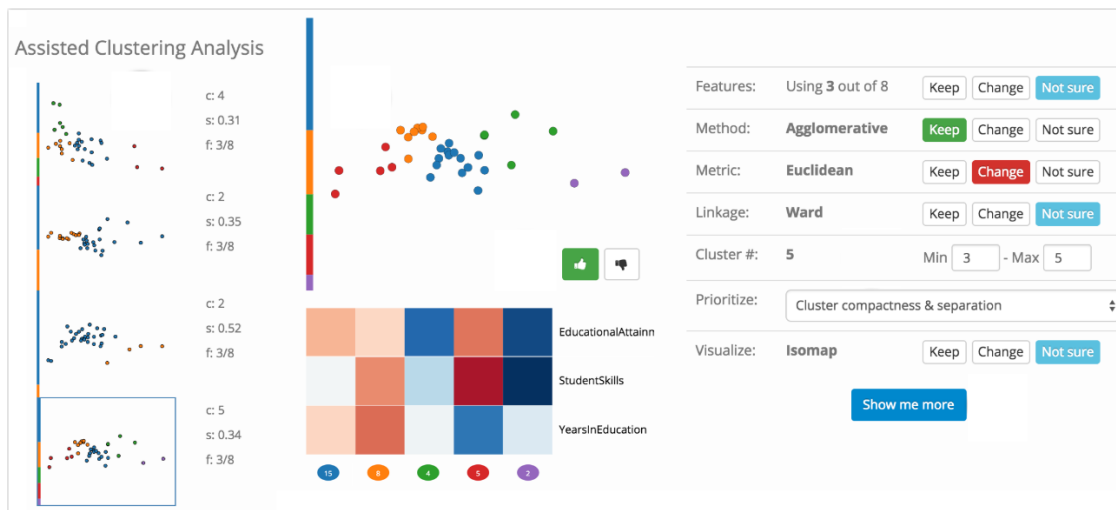


Figure 2.7: The Clustering Tour interface in Clustrophile [51], permitting analysts to explore possible clustering solutions and a dimension-reduced scatter plot projection of the data. © 2019 IEEE.

data exploration [286]. Such opportunities exist in a number of spaces, such as mobile devices [46], wearable headsets (e.g., Oculus Rift), physical spaces (e.g., CAVEs), and motion sensors (e.g., Kinect) [53], and they permit interdisciplinary research collaborations with a variety of fields [217]. Such techniques have long been used for exploring three-dimensional scientific simulations [183, 307], but are now being used more frequently for visualization of abstract data. For example, Coimbra et al. present a variety of interaction techniques for manipulating and gaining insight from 3D scatter plots [73].

2.2.6 Relationship to Clustering

A number of interactive dimension reduction tools also include a clustering component, as a result of the latent relationship that exists when finding dense areas of observations within a dimensionally-reduced space. For example, Clustrophile [51] (Figure 2.7) includes multiple clustering views which include data projection components using a variety of dimension reduction algorithms. Clustering views are provided to supplement projections [284], and

they can also overlay projections, as seen in “Be the Data” [56], iVisClustering [196], and TopicLens [181], among many other tools. As a result, this is a natural transition to a discussion of clustering algorithms and tools.

Chapter 3

Background: Clustering Algorithms and Tools

This background chapter presents a brief survey of clustering algorithms (Section 3.1) and tools (Section 3.2). Hundreds of clustering algorithms have been implemented, each with inherent strengths and weaknesses. The broad collection of approaches in this class of algorithms stems from the notion that a “cluster” is inherently a subjective structure, and as such is difficult to precisely define [118]. While it is generally agreed that a cluster is a region of higher density of observations in a space [144], determining the precise boundaries that determine the constituent observations of a cluster are dependent on chosen parameters. The subjectivity of clustering is demonstrated in Figure 3.1, in which participants were asked to draw clusters in a projection of the States dataset [102]. The clusterings in the left column are all somewhat (but not precisely) similar, each containing five clusters and allowing for outliers that are not assigned to any cluster. However, those presented in the right column are quite different, with a variety of clustering assignments, different numbers of clusters, and few outliers.

There is no single clustering algorithm that suits every clustering problem, but all clustering algorithms manipulate parameters to identify overdense regions of observations within an input space. Therefore, new algorithms or improvements on existing algorithms are often created to solve a single problem, though these new solutions may be applied to future problems



Figure 3.1: Six different user-created clusterings from a States dataset [102] projection. Several of the clusterings (left column) are quite similar, while others (right column) are much more diverse.

where appropriate. As a result, there is no globally optimal clustering algorithm; the best clustering algorithm is problem-specific and often determined experimentally [118]. Surveys of clustering algorithms exist in the literature, which include clustering from the perspectives of machine learning, human-computer interaction, visualization, and statistics [64, 332].

3.1 Clustering Algorithms

A difficulty shared by many clustering algorithms is the computational complexity of locating the optimal solution. For example, finding the optimal clustering assignments for n observations via divisive clustering is $O(2^n)$ [119], while via k -means it is $O(n^{dk+1})$ [163]. Further, the most interesting formulations of the biclustering approach are NP-Complete [241], optimizing any centroid-based method is NP-Hard [130, 308], and there are 2^d different subspaces to consider for subspace clustering in a dataset of d dimensions [190]. As a result, these clustering algorithms make use of heuristics to improve runtime while still providing an approximate solution.

The discussion in this section begins with hierarchical clustering, before proceeding to centroid models, distribution-based models, density models, and subspace clustering technique groups. Following this is a discussion on evaluating the quality of clustering, as well as a brief mention of soft clustering.

3.1.1 Hierarchical Clustering

Hierarchical clustering algorithms come in two primary forms: divisive (top-down) and agglomerative (bottom-up) [256]. The divisive strategy approaches the identification of clusters through iterative partitioning, beginning with a single group and breaking it down

into smaller portions [7]. In contrast, agglomerative algorithms approach clustering through iterative aggregation, beginning with every item in its own group and joining groups together [253]. The structure of the resulting clustering hierarchy is often depicted in a dendrogram, an organized tree diagram which indicates the separation of clusters from each other, often also including the measure of dissimilarity necessary to cause a cluster separation along one axis.

In order to determine whether a cluster partition or aggregation is necessary, a distance metric is necessary (see Section 2.1.7 for a summary of these). The challenge of measuring the distance between two clusters is increased by the fact that clusters are a distributed construct rather than a single observation. As a result, the measured distance is dependent upon both the choice of metric and the choice of linkage criterion. Common choices for linkage criteria include single linkage (a measure of the closest distance between observations from each cluster, nearest neighbor), complete linkage (the distance between the most-separated observations from each cluster, furthest neighbor), and average linkage (the mean distance between observations of each cluster), though other strategies also exist in the literature [289]. The linkage criterion selection can have a dramatic impact upon the cluster hierarchy and hence the dendrogram, as seen in Figure 3.2.

3.1.2 Centroid Models

Perhaps the most common clustering method is k -means [167], a centroid model for clustering which partitions a dataset into k clusters according to a distance between each observation and the nearest cluster centroid. Depending upon the implementation, these cluster centroids may be observations in the original dataset (often referred to as k -medoids [178]), but more often they represent the average of all observations that have been assigned to that cluster

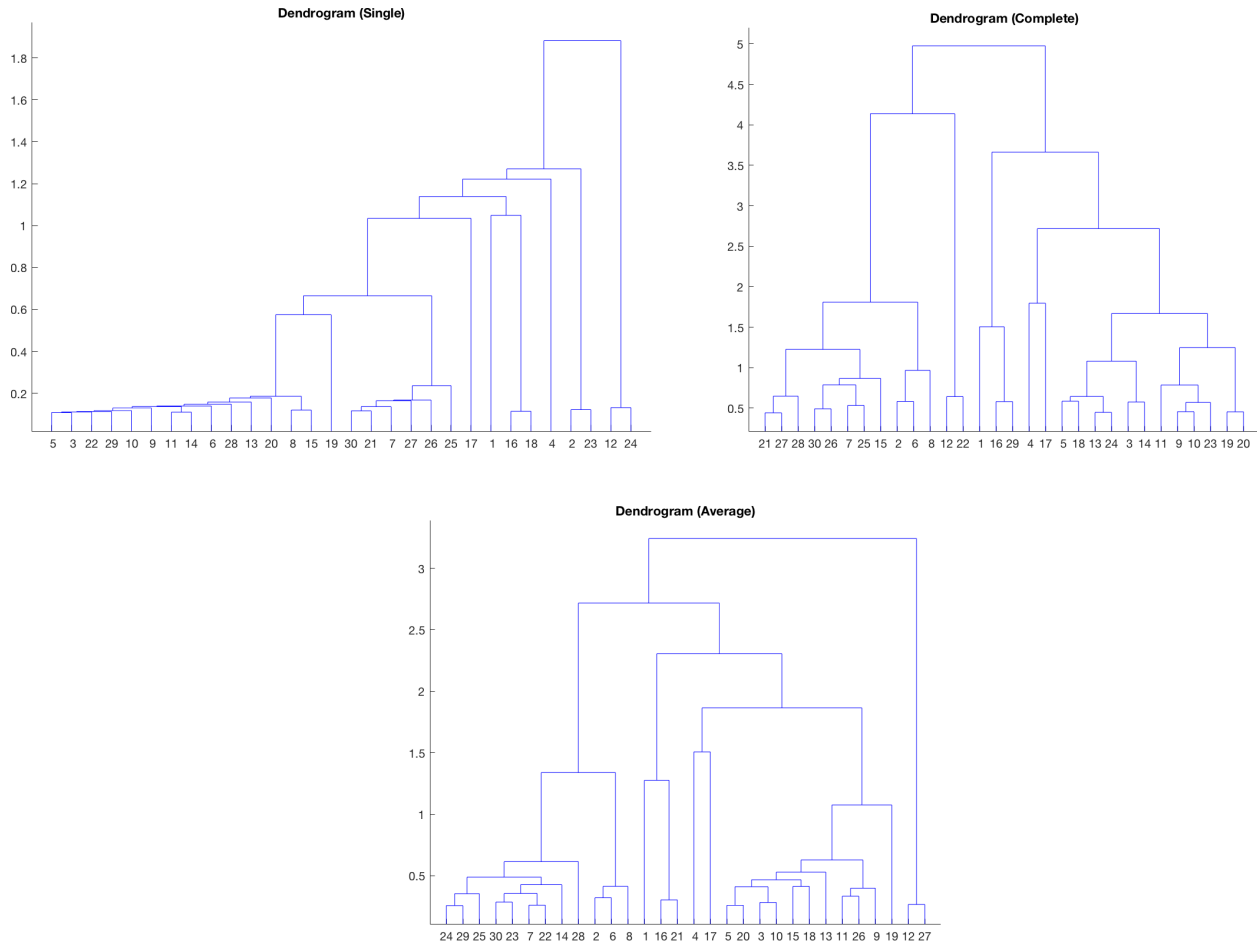


Figure 3.2: Three dendrograms produced using MATLAB from the same input synthetic dataset using single, complete, and average linkage criteria.

in the current iteration. After a solution to the chosen algorithm has converged, the space is effectively partitioned into a Voronoi tessellation with the cluster centroids serving as the seed points [20].

The standard algorithm for approaching k -means is Lloyd's algorithm [209], which refines cluster assignments in an iterative process of assigning each observation to the nearest cluster and the calculating a new centroid based on those assignments, continuing to iterate until no assignments have changed. Lloyd's algorithm reaches a local minimum with a computational complexity of $O(nkdi)$ for n observations, k clusters, d dimensions, and i iterations. The

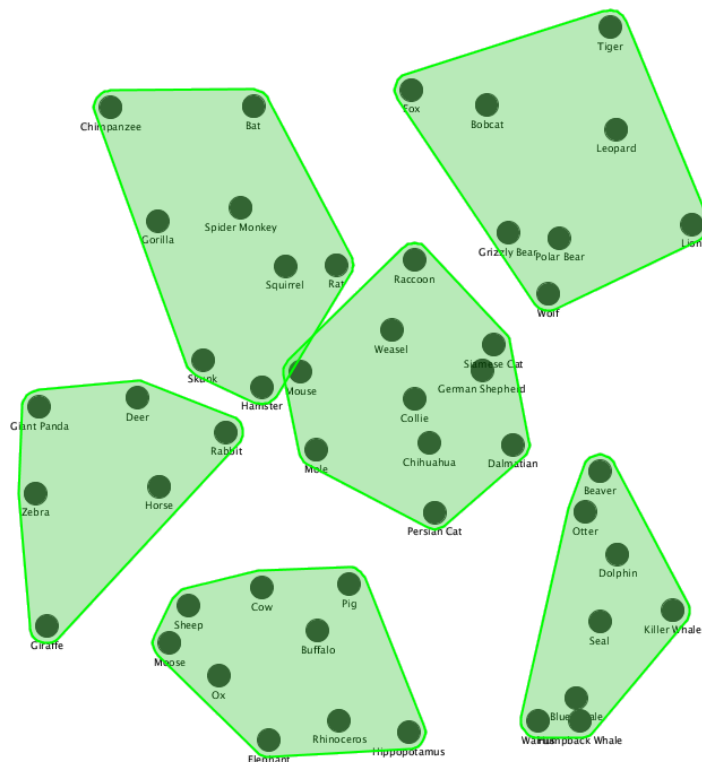


Figure 3.3: The k -means algorithm finds six clusters of animals in the Animals dataset [195], as shown by the Castor system [317].

k -means algorithm has been extended to support a variety of tasks, including weighted clustering [162], hierarchical clustering [242], textual data [86], and constrained clustering [312]. A number of k -means variants are discussed in detail by Cordeiro de Amorim and Mirkin [77]. A major limitation of k -means and related centroid-based approaches to clustering is that they can only find clusters with convex shapes (Figure 3.3). The k -means algorithm also requires input parameter k for the number of clusters to create, presenting an additional complication in generating the best set of clusters with its heuristic approach. Several solutions to determine the most appropriate k value are used, such as the elbow method [296]. However, the elbow method can often be imprecise. For example, Figure 3.4 shows an example with a synthetic dataset in which the elbow method reports 9 clusters but the human eye can easily spot 10.

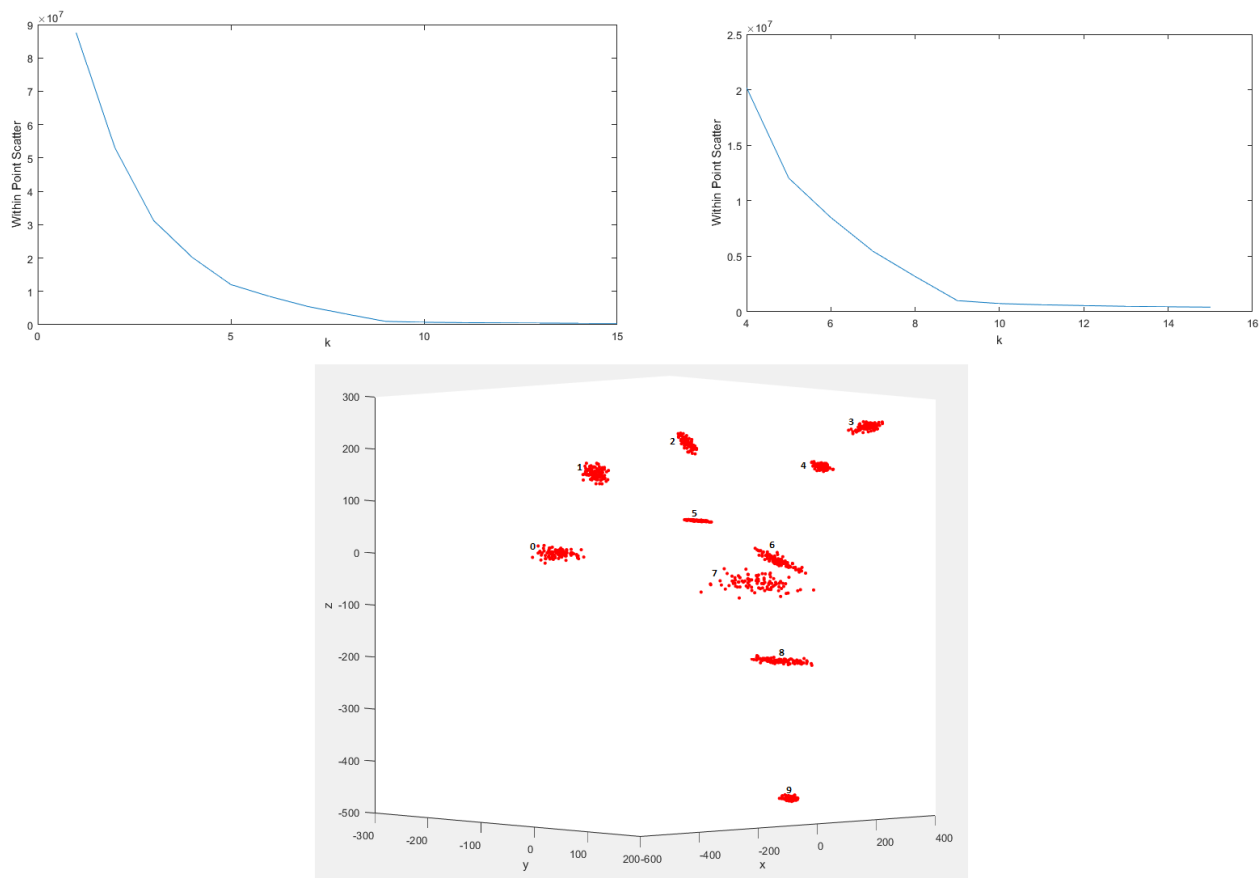


Figure 3.4: Although the elbow method shows a total of 9 clusters in this synthetic dataset, plotting the data in three dimensions shows that there are in fact 10 clusters. Clusters #6 and #7 are separable to humans, but separating them computationally does not substantially reduce the total intra-cluster variance.

3.1.3 Distribution-Based Models

Because clusters can be thought of as dense regions within a space, they can be approximately modeled by probability distributions, where each cluster in the high-dimensional space is a set of observations that belong to the same distribution [220]. For example, under the Gaussian mixture model (GMM) approach, the input dataset is modeled with a set of Gaussian distributions that are iterative optimized to best fit the observations via the expectation-maximization algorithm. As the algorithm iterates, the distributions converge to a local minimum [22]. Overfitting the observations can be an issue in distribution-based

approaches, similar to assigning every observation to its own cluster in k -means to achieve zero error. As such, constraints often need to be placed on model complexity [52].

A common clustering method used in statistics is the Dirichlet process mixture model (DPMM) [33]. Unlike k -means, DPMMs learn the number of clusters dynamically, creating new clusters and closing old ones as the algorithm proceeds. It is not without drawbacks, however. The DPMM requires specification of a probability model for the observations in each cluster, which in turn introduces its own difficulties. The algorithm also scales more poorly than k -means with additional data, especially if the model parameters are estimated with Markov chain Monte Carlo.

3.1.4 Density Models

The goal of density-based clustering is to identify areas of high observation density within the input space, treating observations between these dense regions as noise or border points [189]. Similar to the linkage criterion necessary in hierarchical clustering, density-based techniques require a density criterion that counts the number of other observations within a fixed distance from the observation under consideration. All observations that interconnect within these density computations are treated as a cluster. These density models can find clusters of many shapes and do not require an input k parameter, but they do require a distance criterion to determine the cluster border, which also requires tuning.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [117] is one of the more commonly-used density-based clustering algorithms, operating by grouping together closely-packed observations and marking observations in low-density regions as outliers. A cluster under DBSCAN satisfies the properties that all observations in the cluster are mutually density-connected and any observation reachable from any observation in the cluster

is also a part of the cluster. The algorithm iteratively selects an observation P that has not been labeled as part of an existing cluster, performs a range query to determine all other observations within a distance of ϵ from P , and makes a determination as to whether P should be labeled as noise or as the seed of a new cluster based on the number of neighbors detected. In the case that it is a part of a new cluster, the neighbors are then added to a seed set that grows as new neighbors-of-neighbors and beyond are considered for membership.

The OPTICS (Ordering Points To Identify the Clustering Structure) algorithm [18] is an extension of DBSCAN which orders observations to locate nearest neighbor sequences. These sequences become interconnected in a spanning tree, from which clusters can be extracted by setting an ϵ threshold for largest possible neighbor edges to consider. A similar algorithm, Density-Link-Clustering (DeLi-Clu) [3], removes the need for the ϵ parameter through an additional indexing step.

3.1.5 Subspace Clustering

Clusters are intended to group related observations, but when the number of dimensions becomes too great, some dimensions may not be meaningful for a given cluster. It is further likely that some dimensions are correlated, and as such do not provide new information to the clustering process. The goal of subspace clustering is to identify a smaller, relevant set of dimensions that can be used to structure a particular cluster [190]. The CLIQUE algorithm [6] operates by identifying low-dimensional subspaces and then combining them to build a higher-dimensional subspace, allowing clusters that have been identified in smaller subspaces to persist in larger ones. SUBCLU [173] is a subspace clustering method that uses a density-based clustering approach borrowed from DBSCAN, again building from lower-dimension to high-dimensional subspaces.

Biclustering [143, 223] approaches the subspace clustering problem somewhat differently, with the earliest successful approaches using a variance-based strategy to identify subsets of observations with similar properties in a subset of dimensions [58]. The goal of these biclustering algorithms is to simultaneously cluster both observations and dimensions in order to identify pockets of similar behavior within a larger dataset. Later approaches build on the Cheng and Church algorithm to incorporate graph partitioning [85] as well as concepts from information theory [21, 87].

3.1.6 Evaluating Cluster Quality

A number of methods exist for assessing the quality of clustering, each of which has drawbacks resembling the subjective quality of human evaluation [122]. Thus, there is no single best method to evaluate clustering, just as there is not necessarily a best clustering algorithm [244]. Techniques for computationally determining the quality of clustering are often divided into *internal* and *external* validation methods.

Internal validation methods create a scoring scale that seeks high similarity within clusters and low similarity between clusters. Such unsupervised evaluation techniques are once again based on the choice of distance metric and linkage criterion used to ultimately judge the “similarity” of a pair of clusters. One of the first internal validation methods is the Dunn Index [105], which computes the ratio between the minimum inter-cluster distance and the maximum intra-cluster distance, with the aim of providing the best score to clusterings with dense, well-separated clusters:

$$DI_m = \frac{\min_{1 \leq i < j \leq K} \delta(C_i, C_j)}{\max_{1 \leq k \leq K} \Delta_k} \quad (3.1)$$

Here, $\delta(C_i, C_j)$ represents the distance between the centroids of clusters i and j , while Δ_k

represents the intra-cluster distance of cluster k . All K clusters are examined to find the smallest inter-cluster and largest intra-cluster distances.

A similar approach is found in the Davies-Bouldin Index [81], again creating a ratio of the intra-cluster distance and the inter-cluster distance, but now including each observation and centroid in the ratio rather than simply computing the minimum distance between clusters and the maximum distance within a cluster:

$$DB = \frac{1}{K} \sum_{i=1}^n \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{\delta(c_i, c_j)} \right) \quad (3.2)$$

Here, σ_x represents the average distance of all elements in cluster x to centroid c_x , and $\delta(c_i, c_j)$ is again the distance between centroids of clusters i and j .

More recently, the Silhouette Coefficient [258] has been introduced, which compares the average distance of each observation to other members of its assigned cluster and contrasts this with the average distance to observations in all other clusters, producing a score in the range $[-1, 1]$ for each observation. The mean score of each observation then provides a measure for the quality of clustering across the entire dataset. While these three techniques are commonly seen, a number of other methods have been proposed, including the Gap Statistic [298], the S_Dbw Index [140], the PBM Index [236], and the Xie-Beni Index [330].

External validation methods use an external ground truth to evaluate the quality of the clustering. These supervised evaluation techniques rely on class labels often created by expert humans [244]. For example, the Rand Index [251] judges the number of correct decisions made by a clustering algorithm, a ratio of the number of true positives and true negatives with all clustering assignments. The F-measure [216] improves on the Rand Index by permitting false positives and false negatives to have differently-weighted effects on the score through the introduction of precision and recall measures. Further variations of the

Rand Index ratio formulation are found in the Dice Index [88] and the Fowlkes-Mallows Index [125].

A related problem to evaluating cluster quality is simply determining the number of clusters that exist within a dataset. The ideal number of clusters represents a balance between permitting an amount of error in the clustering and limiting the overall number of clusters. In other words, increasing k without bound will eventually result in zero error when each observation is assigned to its own cluster, but a substantially smaller number of clusters may be found that still has a small amount of misclassification error. This tradeoff is the basis behind the elbow method [296], searching for an “elbow” in a plot of number of clusters versus percentage of variance explained. At such an inflection point, adding an additional cluster does not substantially reduce the amount of error in the clustering assignments. Information criteria such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC) and Deviance Information Criterion (DIC) can be used to further evaluate the quality of introducing another clustering subdivision. This has been implemented in the k -means extension X -means [242]. Other clustering determinations make use of information-theoretic measures such as rate distortion theory [288] and feature rescaling [82].

3.1.7 Multiple Cluster Assignment

Soft clustering, also frequently referred to as fuzzy clustering, permits each observation to belong to multiple clusters. This multiple assignment could be probabilistic, with observations belonging to a individual clusters with varying degrees of probability, or it could be deterministic, with observations assigned to multiple, cross-cutting clusters [333, 339]. For example, the Fuzzy C-means Clustering algorithm [30, 104] is a fuzzy extension of the k -means algorithm, in which the centroid of a cluster is now computed as the mean of all

observations weighted by their probability of belonging to the cluster. When modeling a dataset with distribution-based approaches like GMMs and DPMMs, several models may overlap, and so rather than return a hard clustering assignment, it gives each observation a probability of belonging to any given cluster. A hard clustering assignment can still be generated by these techniques by assigning an observation to the cluster with the greatest probability. Fuzzy clustering assignments also require different validation measures than hard clustering assignments [131, 330].

3.2 Interactive Clustering Tools

Simply presenting an analyst with a clustering visualization may be sufficient for the analyst to perform tasks such as identifying clusters, seeing relative positions of clusters, determining cluster structure, finding anomalies, or finding correlations. However, Malone shows that interactively categorizing information is an important factor in organization to improve the cognition of data [213]. Interactive clustering serves several purposes for the analysis of data, depending on the goals of the analyst. Typically, the analyst wishes to find the clustering assignment that best suits their current search strategy or supports their targeted conclusions [16, 212]. Systems such as SOPHIA provide analysts with support for exploratory search and retrieval of documents, in this case for medical documents [91]. The goal of an ideal interactive clustering system is to understand these analysts and adapt the clustering to suit their intent [64, 137, 282]. Some systems provide analysts with options, displaying multiple clustering results and allowing the analyst to choose the best solution [108]. Still others aim to highlight interesting data automatically for the analyst, guiding their exploration to regions of the data [17, 36]. Interactive topic modeling allows analysts to see groups of documents based on common topics of interest [108, 154, 160, 181].

We begin the discussion of interactive clustering with a discussion the use of clustering algorithms for data exploration, focusing on the role of analysts gaining insight about their data via interactive clustering. This is followed by a discussion of human-in-the-loop clustering systems, in which an analyst provides feedback to a semi-automated clustering algorithm to improve the current state of the clustering assignments. We describe the goals and applications of several such systems. We next present a discussion on methods by which a clustering algorithm can get feedback from an analyst. Finally, we discuss several studies performed to judge the effectiveness of interactive clustering.

3.2.1 Interactive Clustering for Data Insight

Exploratory data analysis is the often visual approach to evaluating, analyzing, and summarizing a dataset [302, 337]. With this approach, an analyst can identify and begin to explain data characteristics such as outliers and extrema, distributions, trends, and patterns. Most systems designed for exploratory data analysis afford one or more interactions to allow the analyst to obtain more information about an observation or about a cluster in order to better understand the layout and grouping of the observations. These interactions are almost universally details-on-demand via mouseover [63, 181, 196, 317], though there are other methods that support the acquisition of contextual information.

For example, Termite [65] allows analysts to click on a term to view its distribution across the entire dataset, as well as to click on a topic to view its representative documents. TopicLens [181] provides a resizable mouseover lens that dynamically divides the overlaid subset of observations within the lens into subclusters, enabling an analyst to see finer-grained structure among the observations. Analysts are also able to filter contextual information to only the most salient observations [65]. Clustrophile allows an analyst to manipulate clus-

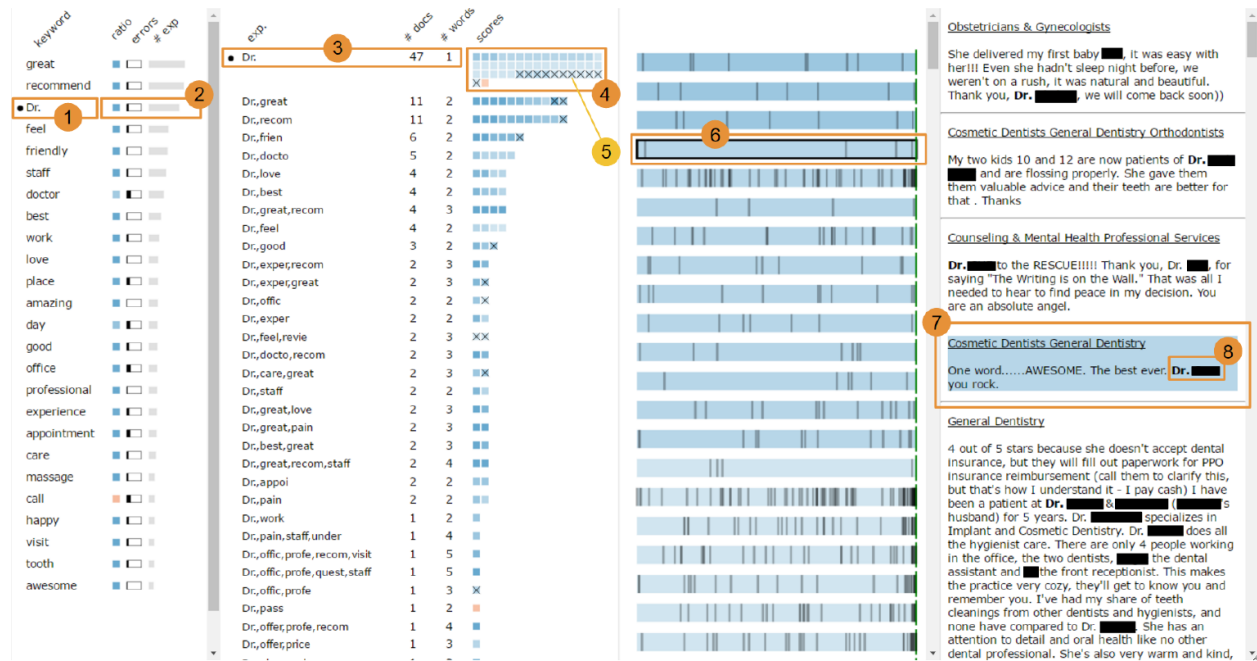


Figure 3.5: Rivelo [290] provides analysts with interactive methods for understanding classifier decisions. Included under Fair Use, 2019.

tering parameters, permitting the rapid exploration of a variety of discrete and continuous clusters and providing evidence to connect clustering instances to data dimensions [83].

Beyond simply identifying clusters, analysts often also wish to understand the cause of clustering, looking for the components of the data and algorithm that are responsible for creating the current view of the data. This is often made difficult when the algorithm or the parameters are hidden from the user. A number of tools have thus been developed to open this metaphorical “black box” and afford analysts with additional understanding of the visualization generation process.

For example, Rivelo [290] (Figure 3.5) provides an interface to explain the causes of classifier predictions, permitting an analyst to interactively explore model-agnostic explanations at the observation level. Analysts can interactively probe the high-dimensional data space in order to detect the features judged to be relevant to the current set of predictions. Not limited to

clustering, Prospector [187] allows analysts to see the relationship between dimensions and the predictions generated by a number of machine learning models. This tool also provides interactions for detailed inspection, allowing analysts to understand how and why individual the properties of observations were predicted by the underlying models.

Analysts can also supply contextual information for their own use as they explore the data, such as labeling clusters [196]. Such contextual information is not limited to the labels and contents of observations and clusters. Some systems supply cluster similarity information [196], as well as sorting [65, 196, 221] and coloring [221] mechanisms to support additional tasks like characterizing distributions. No change is made to clustering assignments or observation layout with any of these interactions; instead, these interactions are simply providing the analyst with contextual information about the observations and clusters.

3.2.2 Human-in-the-Loop Clustering

Analysts often wish to delve deeper than merely inspecting observations and clusters. When exploring a large dataset, an analyst may only be interested in a subset of the dimensions, or potentially interested in a weighted view of the dimensions. Because high-dimensional datasets can be visualized in a variety of projections dependent upon the selected dimensions, giving the analyst the ability to choose between or even generate such projections is important. Human-in-the-loop systems provide a measure of control over the algorithms and its parameters to a human, allowing them to manipulate the results generated by the machine learning module. With respect to clustering, analysts can provide feedback regarding the quality of the groups or the observations that they expect to be contained within the groups, incrementally training the models to reflect their clustering exploration interests.

A number of tools have been developed to permit analysts to create their own groupings

within an interface to use as feedback to the underlying models [70, 155]. For example, desJardins et al. introduced their Interactive Visual Clustering (IVC) technique, which continually generates new observation positions and hence new clusters in the display as the analyst provides feedback by repositioning the observations [84]. They make use of a constrained clustering algorithm which combines attribute information with analyst feedback, generating new edges within the graph to support the new layout. Yue et al. take this technique a step further, analyzing the groupings created by a user population to understand the variability of similarity functions and thereby creating a collaborative clustering model [338]. They demonstrate that their approach produces more effective user models than metric learning and non-interactive clustering systems.

In addition to simply creating groups, these interactive clustering techniques can also be used for systems to generate recommendations. This can act both as a method to speed up classification by approving the machine-generated recommendations, and also permits an analyst to perform a verification check on the underlying model based on whether or not the recommendations are useful. Two representative systems for this approach are ReGroup [13] and iCluster [101] (Figure 3.6). ReGroup assists users in creating custom groups in online social networks by learning the characteristics of the members of the group. From there, the system has the ability to suggest additional members, facilitating the growth of the group. iCluster operates similarly but with a documents use case rather than user profiles. In both cases, the underlying learning models update over time when provided with additional interactions and user feedback.

The properties that define these groups effectively act as labels for the group members. As a result, a number of interactive labeling systems have followed this human-in-the-loop clustering approach. The Exploratory Labeling Assistant from Felix et al. [123] learns an appropriate set of labels interactively, combining the problems of interactive clustering and

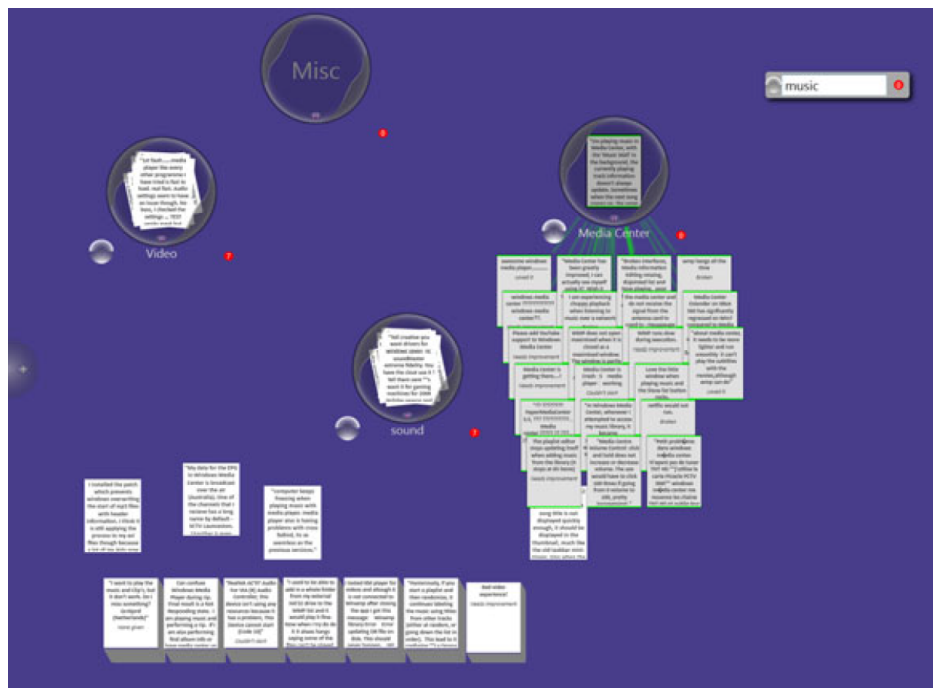


Figure 3.6: iCluster [101] provides an interface for users to interactively cluster documents, and also suggests additional documents that might fit into existing clusters. Included under Fair Use, 2019.

topic modeling in the same system. Further, analysts can specify rules to better couple labels and documents. Raghavan et al. support text classification using support vector machines, taking incompletely-labeled data and allowing a human to iteratively refine the labels of instances for future label assignments [250]. Yimam et al. follow a similar approach for annotating biomedical entities and their relationships, training a machine learning model on a existing set of annotations that are interactively refined by analysts [335]. The system can then recommend these labels to future samples that require annotation.

3.2.3 Getting Feedback

These human-in-the-loop clustering systems are dependent on receiving feedback from the analyst, but there are a variety of ways in which the system can acquire this feedback.

In the simplest case, the analyst can directly update parameters of the clustering model. Many systems also provide analysts with standard GUI widgets such as dropdown menus, slider bars, and checkboxes to alter layout and clustering parameters in the visualization. In ClustVis [221], these parametric interactions allow an analyst to alter the clustering method, linkage method, and the sorting order of the rows and columns in the matrix independently. ClustVis also allows an analyst to change which principal components are used for the axes in its scatter plot view. UTOPIAN [63] uses these controls in a sidebar to enable an analyst to modify both the dimension reduction and clustering algorithm parameters, as well as in a popup to alter the term importances that define the clusters. iVisClustering [196] also provides parameter sliders to manipulate the cluster algorithm parameters, allowing the analyst to directly adjust cluster assignments. The end-user feature labeling approach presented by Wong et al. allows analysts to directly select the features judged to be most responsible for assignment to a cluster or label, including the selection of text from documents [329]. Other systems use a constraint-based approach, iteratively incorporating must-group or must-not-group constraints that the clustering algorithm attempts to satisfy [72].

Some exploratory data analysis techniques also give analysts the ability to directly manipulate the observations and clusters, thereby affording additional tasks. For example, observations can be selected and dragged to repositioning them within clusters or to relocate them to other clusters [25, 70, 74, 317], referred to by Dubey et al. as “Assignment Feedback” [103]. Through such interactions, analysts can supply must-link and cannot-link constraints to clustering solutions [42, 160].

Further, an analyst can be given the ability to directly adjust the number of clusters created, or to directly modify parameters that control those clusters such as a distance threshold [36, 91, 137, 212, 282]. Systems can also support direct interactions with the clusters, such as creating clusters [196, 317], merging and splitting clusters [36, 63, 154, 196], removing

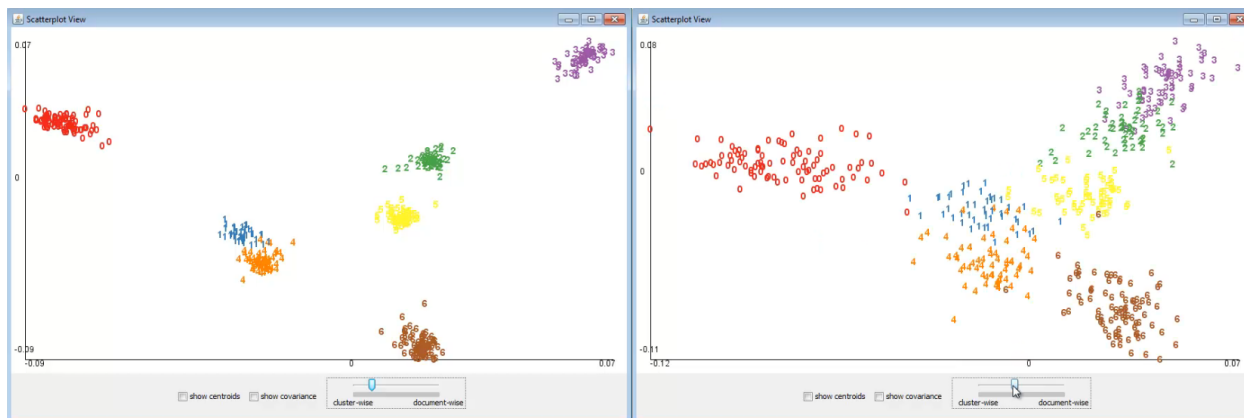


Figure 3.7: The iVisClustering system [196] supports a number of cluster operations, including joining, splitting, creating, and removing. Included under Fair Use, 2019.

clusters [9, 25, 74, 137, 196, 212], and hiding and expanding clusters [16, 25, 36, 91].

Each of these interaction techniques provides a demonstration that algorithms can be created which will re-cluster all data based on how the analyst changes clusters and the memberships of their observations. Effectively, the system will tune clustering parameters on behalf of the analyst. Similar techniques have been used to interactively train classifiers to support image search [12] and to identify interesting projections [26].

Rather than taking an interactive approach, a number of systems make use of active learning to get feedback from users by explicitly asking for feedback when necessary [24, 161, 331, 341]. For example, Druck et al. built a prototype interface that asked analysts to browse groups of related features, selecting the group that they judge to be the best label [100].

3.2.4 Studies

Interactive clustering is only useful as long as the analysts are providing useful feedback. However, a study by Lee et al. found that there is a disconnect between the interactions that non-expert users want and those that human-in-the-loop topic modeling systems sup-

port [198]. These non-experts are not looking to interactively construct groups of documents to communicate with the system; rather, they would rather have operations like adding and removing words from topics, changing the word order, and splitting topics. While systems like UTOPIAN [63] and iVisClustering [196] can support these operations via manually updating the weights applied to words and topics, such interactions are not as intuitive to non-expert users who would prefer a simple button click.

Similarly, Hu et al. performed a study in which participants were asked to group a document collection into clusters according to their own understanding [159]. They found that different users have their own personalized organizational schema that are important to them, but that this organization changes over time. Despite the varying organizational criteria from users, the study also found that semi-supervised clustering with noisy user input still often outperforms transitional unsupervised clustering approaches.

Chuang and Hsu examined interactive clustering approaches in the fields of machine learning, human-computer interaction, visualization, and statistics, and came to the same conclusion: existing techniques are often unsatisfactory [64]. However, rather than arguing from a human-centric standpoint, they feel that current approaches and designs fall short because of insufficient statistical modeling capabilities. To address this, they identify five algorithmic characteristics that are necessary to support effective human-in-the-loop interactive clustering: “(1) iterative, (2) multi-objective, (3) local updates that can operate on (4) any initial clustering, and (5) a dynamic set of features.”

3.3 Future Work

Both this chapter and the previous chapter have detailed the immense number of dimension reduction and clustering algorithms that exist in the literature. However, a number of

these algorithms have not yet been integrated into the systems that follow the discussion of algorithms. Interactive applications that make use of dimension reduction often are limited to simpler techniques such as PCA, MDS, and t-SNE, while those that support interactive clustering often make use of LDA topic modeling and k -means. Sections 2.1 and 3.1 enumerate a variety of techniques to support future research endeavors for the development of interactive visualization tools.

Chapter 4

Dimension Reduction and Clustering Projections

While dimension reduction algorithms and clustering algorithms have been implemented together in a number of visualization systems, these algorithms often operate independently and in parallel. In other words, each algorithm supports some analysis component in the system without the influence of the other algorithm: perhaps a collection of observations are clustered, but the clustering output has limited or no effect on the layout of the observations. Alternatively, a change to the spatialization may perceptually imply the need for a change to the cluster assignment, but no update to the cluster assignment may occur.

Exploring the connection between dimension reduction and clustering algorithms leads to several natural research questions. If the data separates into implicit clusters, and the analyst sees advantages in the creation of these implicit clusters, can we appropriately support explicit cluster definitions so that the dimension reduction and clustering algorithms support each other rather than conflict with each other (or simply do not interact with each other)? If so, how should we define and visualize with observations and clusters in a dimension-reduced projection? And finally, is there a difference between how analysts interpret and interact with low-dimensional clusters as opposed to high-dimensional clusters?

This chapter explores initial steps to address these questions. In particular, it includes the following contributions:

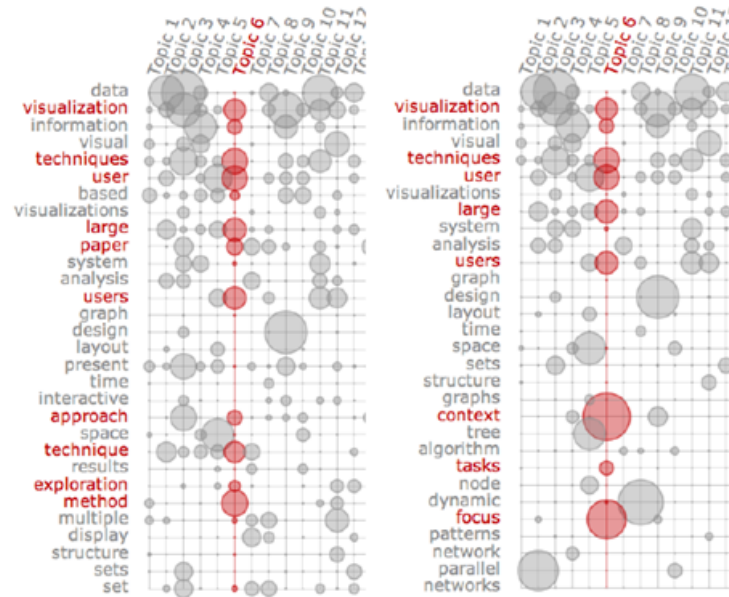


Figure 4.1: The Termite system [65] supports the task of identifying clusters spatially and seeing the relative positions of clusters through a matrix view. Included under Fair Use, 2019.

1. An overview of combining dimension reduction and clustering techniques into a visualization system, including a discussion of tasks and visualizations.
2. A discussion of the design decisions that must be addressed when creating a visualization system that combines dimension reduction and clustering algorithms.

4.1 Tasks

In this section, we discuss the types of tasks that clustering supports and their connection to the low-level analysis tasks by Amar et al. [11]. These tasks directly support steps in the Sensemaking Process [248], and therefore reflect more cognitive benefits of clustering rather than computational. However, a benefit that is reflected both cognitively and computationally is scalability. By clustering observations, an analyst can perform sensemaking tasks with bigger datasets while still retaining the ability to interpret the visualization.

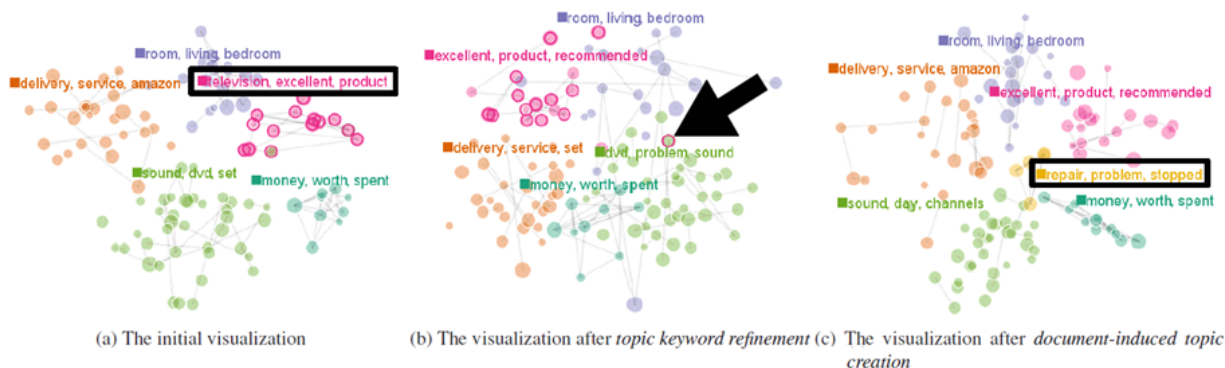


Figure 4.2: The UTOPIAN system [63] supports cluster labeling for exploration and to support better understanding of the data by showing which terms best describe a given cluster. © 2013 IEEE.

To begin our evaluation of the exploratory data analysis tasks supported by clustering, we consider which cluster-centric tasks are most or least commonly supported amongst exploratory techniques that leverage clustering of high-dimensional data. For example, identifying clusters, seeing their relative positions, and determining cluster structure are clearly the most common tasks supported [9, 63, 74, 129, 181, 196, 211, 265, 317], as they should be for tools such as these which are designed to provide insights into clusters of a dataset. Figure 4.1 highlights the Termite system, showing this capability.

In comparison, support for tasks to explore the data are more varied, with changing cluster membership of observations being the most common task of this type supported [74, 196, 317]. The next most commonly supported task in this category is creating or removing clusters [74, 196]. Although direct support for repositioning clusters in the projection space isn't common [74], this is more often indirectly supported through other tasks, such as manipulating a parameter of the clustering algorithm [196, 317]. Understanding the data via the tasks of labeling clusters [63, 129, 181, 221] is a less commonly supported task, though the UTOPIAN system demonstrates this clearly in Figure 4.2.

For exploratory data analysis techniques, clustering also typically implies computing derived values for each cluster (e.g., most salient dimension or topic within each cluster) [63, 65, 181, 196]. Given that finding a high or low extrema attribute within a given cluster is directly supported by these computed derived values, clustering techniques that compute derived values generally support this analysis task as well. Similarly, techniques that show cluster structure directly support characterizing the distribution within that structure [9, 63, 74, 129, 181, 196, 211, 265, 317]. This information can help to uncover correlations between clusters or observations as well as to find anomalies. Other tasks such as retrieving values are commonly supported [9, 63, 65, 196, 317], which may additionally help determine ranges [65]. Filtering [65] and sorting [65, 196, 221] are occasionally supported analysis tasks in exploratory data technique that utilize clustering.

Beyond these general categories of tasks that are supported by clustering, some exploratory data analysis techniques are designed to support specific tasks. Many of these techniques highlight support for refining the clustering results themselves [63, 181, 196, 320], which is a task that is supported mathematically through a combination of visualization and clustering techniques. As another example, iVisClustering [196], Termite [65], ClustVis [221], and UTOPIAN [63] all afford tasks such as understanding which dimensions or terms best describe a given cluster. This task is dimension-centric, as opposed to understanding cluster structure, which is an observation-centric task.

All of the tasks supported by these exploratory data analysis techniques demonstrate how clusters can be leveraged to improve sensemaking. That is, these tasks directly support sensemaking tasks. For example, in Pirolli and Card’s Sensemaking Process, clustering in general supports organizing of tasks (which helps the analyst in the “Evidence File” step) or skimming of the data (proceeding from the “Shoebox” step to the “Evidence File” step). Similarly, clustering provides an overview of the data and imposes structure, which helps the

analyst proceed from the “Evidence File” step to the “Schema” step. Filtering and sorting may assist with these two steps of the sensemaking loop in addition to enabling searching for specific types of data (proceeding from the “External Data Sources” step to the “Shoebox” step). Thus, any additional task supported by an exploratory data analysis technique that leverages clustering, including those described previously, only further enhances its ability to support sensemaking.

4.1.1 Dimension Reduction and Clustering Tasks

This discussion of tasks for dimension reduction and clustering algorithms focuses on exploratory data analysis. When exploring a high-dimensional dataset with dimension-reduced projections, there are an immense number of possible 2D- or 3D-projections that can be generated from the dataset. An analyst should be afforded the ability to explore these alternate projections, as well as the related clusterings in those projections, in order to gain insight from the data.

One method for enabling this exploration is by applying weights to the dimensions in the dataset. Biasing the algorithms towards combinations of dimensions in the dataset enables the creation of projections that are similarly biased towards those dimension combinations. Thus, an analyst can explore clusters and patterns in a projection that is biased towards dimensions X , Y , and Z , and contrast that result with clusters and patterns in a projection biased towards only dimensions U and V , both from the same initial dataset.

When interactively exploring a dataset, dimension reduction tasks (the left columns of Table 4.1) typically relate to position, while clustering tasks (the right columns of Table 4.1) typically relate to grouping. For example, identifying a similarity relationship between two observations based on their separation distance in a projection is a dimension reduction task,

Table 4.1: Sample exploratory data analysis tasks, organized by stage in the data analysis process (rows) and algorithm family (columns).

	Dimension Reduction	Both	Clustering
See the Result	See distribution of observations	See relative positions of observations	Identify clusters of observations
Understand the Result	Measure distances between observations	Identify attribute values of observations	Label clusters Determine cluster structure
Affect the Result	Change distance metric Select different dimensions	Reposition observations in the full space Enhance an existing pattern in the projection	Change cluster membership of observations Create/remove clusters

while positioning two similar observations close together is a clustering task. However, there exists obvious ambiguity even with such basic interactions. When positioning two objects close together to form a cluster, the analyst is also communicating a distance relationship between those observations. Thus, space is overloaded for both grouping and layout interactions, further suggesting a relationship between the dimension reduction and clustering algorithm families. As seen in the selected tasks breakdown in Table 4.1, tasks can often be addressed by only using a dimension reduction algorithm or a clustering algorithm, but there do exist many cases where the interplay between algorithms affects both when a task is performed.

This relationship can be further seen in Brehmer et al. [41], in which ten analysts from six application domains were interviewed with the goal of understanding how analysts explore dimension-reduced data. The end result of this study was a set of five task sequences. Although the authors were focused on analyst interpretations of dimension-reduced data, three of the five resulting task sequences were related to clusters of items revealed in the low-dimensional data projection. Indeed, the “Verify Clusters” task sequence was performed by all ten of their analysts and the “Name Clusters” sequence was performed by eight of the ten analysts. In contrast, the tasks sequences that were not cluster-based were only performed by two (“Name Synthesized Dimensions”) and four (“Map Synthesized to Original Dimensions”) of the ten analysts. These findings suggest that analysts are discretizing these

clusters of observations in dimension-reduced projections. In other words, the dimension reduction algorithm is creating a continuous visual distribution that analysts interpret in discrete segments. Moreover, investigating these clusters within the projection are common goals of user exploration and interaction with datasets.

In addition to investigating clusters in an existing projection, studies have shown that analysts create their own clusters of observations. For example, the “Space to Think” study by Andrews et al. [15] investigated how analysts use large displays to navigate and lay out documents in the sensemaking process [248], and that these clusters occasionally have spatial relationships, both to develop a timeline and to keep similar clusters of documents near to each other spatially. When interviewed about their sensemaking process later, analysts spoke of their documents and clusters both in terms of proximity and in terms of groups, implying that these are similar cognitive processes. The ForceSPIRE [115] and StarSPIRE [38] systems were designed in part from these findings.

Similar behavior was seen in the “Be the Data” system reported by Chen et al. [57], which allows participants to explore a dataset by taking on the role of the observations in a defined physical space. By moving about the space, participants update a dimension-reduced projection. The system is thereby able to learn which dimensions of the dataset are most important to the current “projection” of people. Presented with a collection of animals and their attributes, a group of seventh grade students were posed the question “What makes some animals good to eat?” The students began their exploration of the data by clustering animals into discrete Edible and Inedible clusters. However, the student who embodied the Rat observation did not consider herself a part of either group, noting that rats are normally not edible but are consumed in some cultures. She then positioned herself between the two clusters. This caused the rest of the students to reconsider their distribution, turning their discrete clusters into a continuous distribution of Edibility.

Cluster investigation tasks (the right columns of Table 4.1) come in a number of forms, each of which have some meaning in a dimension-reduced projection. For example, analysts may wish to understand the overall layout of clusters in a projection, explore the proximity of one cluster to another, investigate clusters of clusters and similar structures, the shape of a cluster, and describe outlying clusters versus central clusters. In addition, analysts may be interested in the relationship between clusters and the individual observations in the projection, exploring to which cluster(s) an observation belongs, understanding the properties of observations that are outliers to all clusters, and investigating the properties of a set of observations that form a cluster. There is a mix of distribution and group questions that can be addressed through the combination of both dimension reduction and clustering algorithms.

Adding clusters and clustering interactions to dimension-reduced data can also improve scalability as datasets continue to grow in size [107]. Having the ability to abstract collections of observations into a single cluster that acts as an interaction target enables the ability to place more objects into virtual spaces, useful both for standard monitors and for large display systems.

While the outputs of dimension reduction and clustering algorithms are useful to locate patterns in a dataset, we also benefit from enabling these algorithms to learn from user interactions [109, 157, 199]. By interpreting the semantic meaning of user interactions, each of these algorithms can better enable exploratory data analysis. For example, an analyst may wish to know what model parameters are necessary to create a cluster from observations A , B , and C . By manipulating the projection to form such a cluster and initiating a semi-supervised machine learning routine, the dimension reduction and clustering algorithms can be trained to learn such model parameters and to update the entire projection in response to those new parameters. The new projection may create a new cluster from observations D , E , and F in addition to the analyst-created cluster, a new insight into the dataset. Therefore,

the dimension reduction and clustering algorithms can help both at the beginning of the exploration process by providing a naïve starting point, as well as throughout the exploration process by responding to the interactions of an analyst.

4.2 Coordinating the Algorithms

Another consideration in selecting dimension reduction and clustering algorithms is determining what parameters should be learned and used by each algorithm, as well as what information should be learned by the analyst. Beginning with the analyst, we discussed in the previous subsection that dimension reduction algorithms and clustering algorithms serve similar purposes. However, dimension reduction algorithms are more suited to tasks for pairwise comparisons and similarities between observations, while clustering algorithms are better suited for comparisons involving the recognition and description of groups.

For the algorithms, one obvious design decision is to determine whether or not the dimension reduction algorithm and clustering algorithm should be using the same distance function, or even if they should be using the same set of weights on the dimensions. It is possible for the dimension reduction algorithm and the clustering algorithm to store separate sets of weights, or to use different distance functions entirely.

When considering the semantics of the order of dimension reduction and clustering algorithms, using clustering in high-dimensional space as the first operation makes uncovering clusters the primary semantic role of the system, and hence results in a system designed to support locating and understanding groups in the input data. In contrast, clustering as the second operation in the low-dimensional space after executing a dimensional reduction algorithm results in a clustering algorithm that is merely a secondary aid to the dimension reduction algorithm.

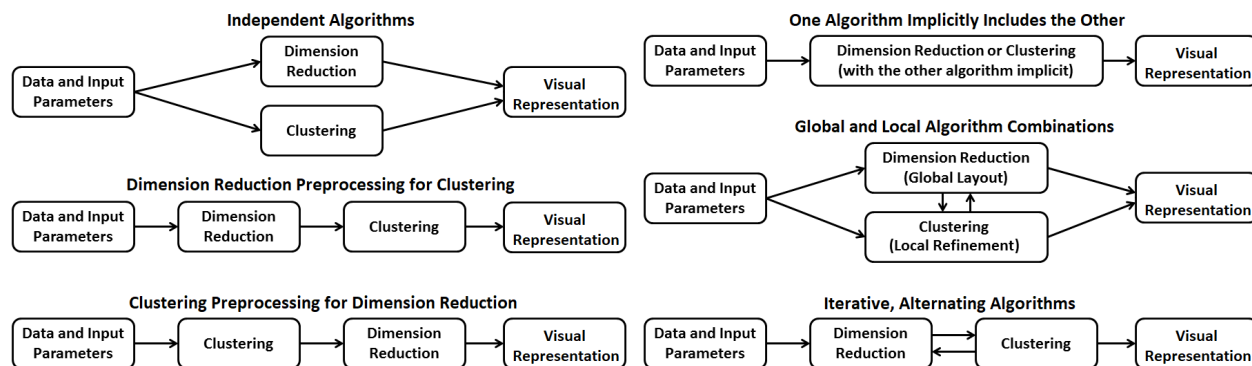


Figure 4.3: Six different options for pipelines depicting combinations of dimension reduction algorithms and clustering algorithms. In each of these pipelines examples, it is implied that each algorithm could use an independent distance function, resulting in more than just these six pipelines. Further, these pipelines represent a single analysis iteration.

An open question is determining whether analysts are cognitively clustering in high- or low-dimensional space. Given that analysts typically form clusters of text documents directly from the text instead of first converting those documents into another form [15], it appears that clustering is performed in the high-dimensional space, at least for textual data. Understanding the clustering process of analysts will lead to better semantic interactions in this dimension reduction and clustering design space, leading to further system interactions such as enabling humans to provide corrections to clustering assignments and hence update dimension reduction algorithm weights and projections. Naturally, it is not possible to coordinate all pairs of dimension reduction and clustering algorithms – some dimension reduction algorithms such as PCA do not rely on distances between observations. Therefore, using the same distance measure between PCA and a clustering algorithm is not possible.

4.2.1 Dimension Reduction and Clustering Combinations

When developing a system that includes both dimension reduction and clustering algorithms, it is important to consider the order in which these algorithms are performed on the data, as the order of these algorithms will generate projections with different semantic meanings.

Figure 4.3 includes six different pipelines that display execution orders and data flows between these algorithms. Each of these pipelines is discussed in the following paragraphs. As the analyst progressively explores the dataset, they may select a different pipeline for each round of exploration, continuing to explore new projections (Figure 4.4).

Independent Algorithms: As discussed previously, many visualization systems incorporate both dimension reduction and clustering algorithms, but these algorithms often execute independently and in parallel so that the output of one algorithm has no effect on the other. This pipeline is highlighted first in Figure 4.3 and was discussed in the iVisClustering [196] example in the Introduction. In this system, topics are computed and assigned as clusters, and a force-directed computation performs the node layout in the spatialization. However, an update to the layout has no effect on the clustering assignments. In addition, computing both algorithms on the high-dimensional data will be more computationally expensive than performing only a single high-dimensional computation.

Dimension Reduction Preprocessing for Clustering: Another possibility is to execute a dimension reduction algorithm on the high-dimensional data, and then pass the low-dimensional projection to the clustering algorithm to determine groups, clustering on the reduced data rather than the source data. This decision may be advantageous because the clustering algorithm can execute faster on a dataset with fewer dimensions, but the outcome may be misleading because the low-dimensional positions of each observation are an approximation of the high-dimensional relationships. Rather than generating clusters of the input data, we generate clusters using data with less information, resulting in potentially misleading cluster assignments. This risk is discussed by Joia et al. [170], noting that distances in the low-dimensional space may be misleading due to projection errors. As a result, what appear to be distinct clusters must be confirmed, as there is no guarantee that these clusters do contain unique content. An example of this pipeline can be seen in Zha et

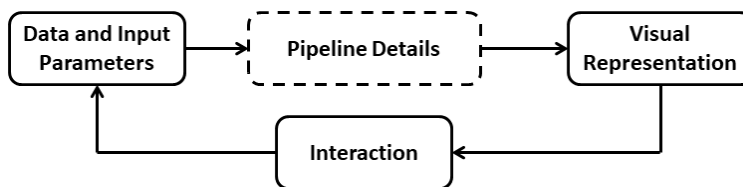


Figure 4.4: Interactions from the analyst will drive additional executions through the pipeline during the data exploration process. The analyst does not need to select the same pipeline on every iteration of the analysis.

al. [340], in which a technique similar to PCA is performed first and followed by k -means on that output. Likewise, Ng et al. [229] propose an algorithm in which the observations are embedded in low-dimensional space such as the eigenspace of the graph Laplacian, and then k -means is applied to that low-dimensional projection. “Be The Data” also creates clusters dynamically based on the current projection [57].

Clustering Preprocessing for Dimension Reduction: The reverse of the previous behavior occurs when the clustering algorithm is the first to execute, and then some information from the clustering output (the cluster assignments, or the locations of the centroids) is used by the dimension reduction algorithm for layout. Now, the clusters represent relationships that exist in the initial data in high-dimensional space. However, the clustering algorithm will take longer to execute due to the additional number of dimensions processed. While less common, some systems do operate in this way. For example, Ding and Li first use k -means clustering to initially generate class labels, followed by LDA dimension reduction for subspace selection [90]. Fuzzy (or soft) clustering introduces a new complexity to this pipeline, as cluster assignments are now a probability distribution rather than a fixed bin assignment. A pipeline of this form can also improve scalability, as the time and space complexity of many dimension reduction algorithms make them infeasible to execute on very large datasets. Clustering observations and then performing a dimension reduction algorithm on those clusters is one solution to this challenge.

One Algorithm Implicitly Includes the Other: Another alternative is to only execute one of the algorithms, either dimension reduction or clustering, and then convert or interpret the output of the executed algorithm as the output for the other algorithm as well. In these cases, the results from one algorithm are structured to fit the objective of the other algorithm, exploiting the mathematical equivalence between these algorithm families discussed briefly in the Introduction. For example, we can codify soft k -means clustering as assigning n observations to k features with some associated weight or probability. Likewise, we can formulate dimension reduction as reducing m features to p features with some associated weight or probability. Therefore, the outcome of soft k -means clustering can be interpreted in terms of dimension reduction by making the k clustering features also represent the p dimension reduction features. A similar argument exists to map the outcome of a dimension reduction algorithm directly to a cluster encoding by executing a dimension reduction algorithm like PCA and binning the output along one of the axes. Perhaps a more straightforward example of this pipeline is the self-organizing map [184], a dimension reduction technique which can be directly interpreted as a set of clusters without any feature transformation. Kriegel et al. [188] present a survey of clustering techniques for high-dimensional data, and include a discussion on subspace clustering algorithms. Such algorithms simultaneously reduce both the number of observations and the number of dimensions in a dataset, in contrast with having a dimension reduction algorithm that reduces the number of dimensions computing separately from a clustering algorithm that reduces the number of observations.

Global and Local Algorithm Combinations: Because dimension reduction algorithms typically take a global view of the overall space while clustering algorithms take a local view [86], another option is to implement a pipeline in which the overall structure of the space is informed by the dimension reduction algorithm while local structures are governed by the clustering algorithm. These algorithms can communicate with each other to converge

towards an optimal layout, but each is responsible for its own aspect of the structure. To further clarify the difference between this pipeline and some of those discussed previously, consider organizing a large collection of documents in a display. One possibility is to place related documents into folders, and then organize the folders in the space. This example reflects the “Clustering Preprocessing” pipeline, as we organize the clusters rather than individual documents. In contrast, the analyst could organize groups of documents in the space, and then select and move those groups with respect to one another. This example affords some additional fuzzy clustering capabilities, as a document that may belong to two or more clusters can be placed between those clusters. Here, the overall layout of the documents can be handled by dimension reduction, while some local structures of similar documents are supported by clustering.

Iterative, Alternating Algorithms: The final pipeline represents a structure where both dimension reduction and clustering are working together in the same overarching algorithm. As k -means is an algorithm that alternates between updating cluster assignments and centroid positions, a third stage can be added for dimension reduction. Ideally, this iterative alternating process will enable dimension reduction and clustering to work in harmony to converge towards a best layout, trying to find the right set of dimensions and a good set of clusters simultaneously while also communicating between the algorithms. This pipeline differs from “One Algorithm Implicitly Includes the Other” in that both algorithms process the data cooperatively, rather than only executing one of the algorithms and using its outcome to present both a projection and a clustering. Since both the dimension reduction algorithm and the clustering algorithm will begin on the high-dimensional data, this pipeline will be among the slowest to converge. Niu et al. [232] provides an example of this pipeline.

This collection of pipelines and examples demonstrates methods for combining dimension reduction and clustering algorithms, but are not without limitations. Even extending these

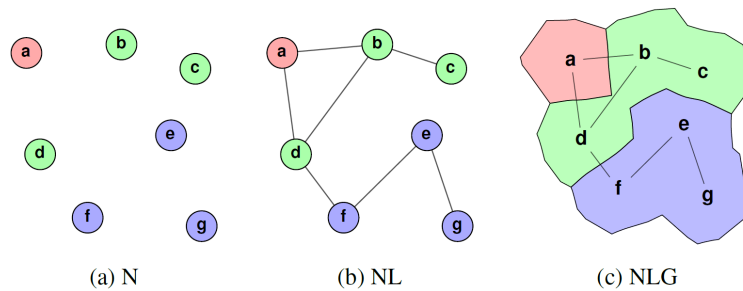


Figure 4.5: Saket et al. [265] evaluate three options for encoding cluster membership, relating each to the effectiveness of performing node- and group-based tasks. © 2014 IEEE.

pipelines with a looping structure to iterate through the dimension reduction and clustering stages is insufficient. To better model this and other similar cognitive processes, we must extend this discussion of algorithms into the realm of visualization and interaction; algorithms alone are insufficient for complex cognition [116].

4.3 Visual Representation

After the algorithms have been selected, the next step is determining how to present the results of the computations to the analyst. In this section, we first discuss common visual representations for dimension-reduced data and clustered data. This is followed by a discussion of potential visual outcomes of the pipelines that were introduced in Sect. 4.2.1.

The variety of methods for visualizing clusters is nearly as broad as the variety of clustering algorithms. Among others, these techniques include encoding cluster membership in color, in position, and in distinctly-separated groups. Such visual encodings assist with tasks such as identifying clusters. In cases where position is used, tasks such as seeing relative positions of clusters, determining cluster structure, finding correlations, and finding anomalies can also be supported. Here, we discuss several example systems that encode cluster membership using these three techniques.

Using color to indicate cluster membership has been demonstrated in a number of tools and prototypes. Saket et al. [265] demonstrated three different methodologies for using color in this manner: coloring nodes, coloring nodes in node-link diagrams, and coloring regions of node-link diagrams. The iVisClustering tool [196] demonstrates this first methodology of using color. Linesets [9] uses colored nodes in node-link diagrams as well as colored links when connecting nodes with the same cluster membership. Lastly, coloring regions of node-link diagrams can be found in GMap [129], which renders a geographic-like map for clusters. Bubble Sets [74] (as shown in Figure 4.6) creates an interesting mixture of the Linesets and GMap visualizations by drawing isocontours around nodes and links to form clusters.

Leveraging color encodings to indicate cluster membership affords preattentive recognition of these clusters [147]. However, there are also cognitive limits to using color to encode cluster membership. For example, the human eye has trouble distinguishing between more than 10 distinct colors in a visualization [279]. As such, color encodings are better suited for visualizations with a small number of clusters.

Position can also be used to encode cluster information using a variety of methods. The Termite system [65] visualizes the relationship between words and topics in a Term-Topic Matrix, a 2D matrix indexed by computed topics on one axis and words on the other. When sorted using their seriation technique, cluster of terms appear as vertical stripes of larger nodes across the words in the matrix. A related technique is seen in the heatmap view of ClustVis [221], which displays dendrograms along the heatmap axes to display the hierarchical cluster structure. The rows and columns of the heatmap are thus positioned based on the ordering of the clustering tree. Cognitively, position is not a preattentive feature, though it can be used to complement closure (discussed next). However, node size is a preattentive feature [147], which is also demonstrated by Termite [65].

A third method for visualizing cluster information is via distinct separation boundaries (i.e.,

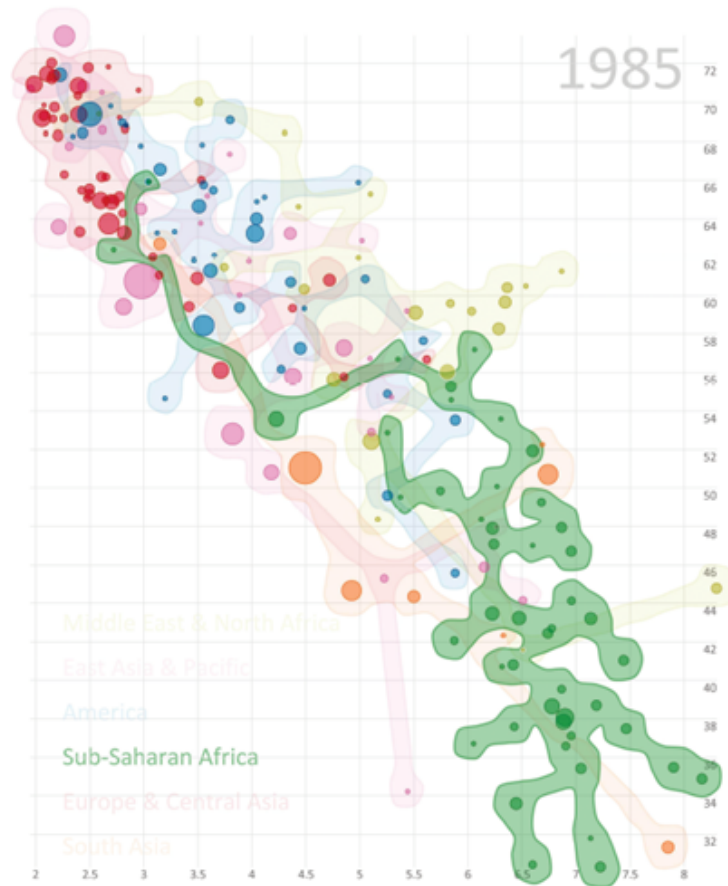


Figure 4.6: Bubble Sets [74] uses the preattentive closure property to display distinct groupings of data, with a clear delineation between what belongs to the cluster and what does not. © 2009 IEEE.

closure), which is another preattentive feature that can be exploited [147] for fast cognitive recognition of clusters. The scatter plot view from ClustVis [221] contains clear boundaries between clusters by drawing ellipses encompassing the observations categorized within each cluster, and the space-filling group encoding studied by Saket et al. [265] has similar, clearly-delineated regions.

A visualization is not limited to choosing only one technique; dual-encoding [145] is a frequently used technique to reinforce cluster membership. For example, both TopicLens [181] (as shown in Figure 4.7) and UTOPIAN [63] use position (via modified versions of t-

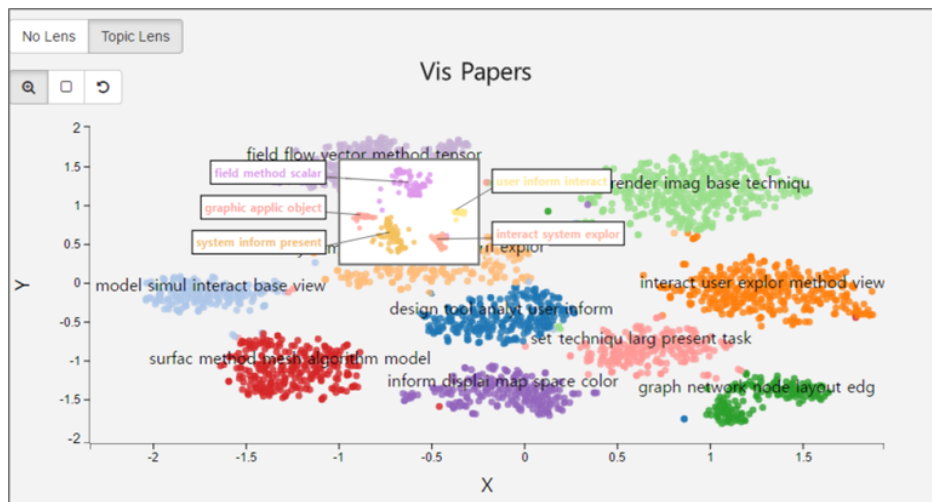


Figure 4.7: TopicLens [181] takes a dual-encoding approach to visualizing clusters, combining color and position. © 2017 IEEE.

SNE [211]) and color to encode clusters of documents.

In addition to cognitive benefits, there are computational benefits to clustering in visualizations as well, particularly as the datasets grow large. For example, consider a visualization system that is given millions of observations to visualize. Visualizing all observations will be computationally slow and cognitively overwhelming. Clustering observations and only visualizing the clusters at some level in the hierarchy reduces the workload on both the computer and the visual system of the analyst. An implementation of this strategy can be found in ASK-GraphView [1] (see Figure 4.8).

However, a major trade-off in only visualizing the clusters is that tasks such as determining cluster structure [320], determining range, or characterizing distribution [11] are no longer readily supported through the visualization. Thus, creating such a visualization imposes a cognitive limitation for the analyst in addition to the aforementioned cognitive benefit.

Another limitation is that visualizations must have a default initial display of the observations and clusters. Presenting an analyst with this initial projection could produce a



Figure 4.8: ASK-GraphView [1] uses clustering to enable efficient exploration of very large networks. © 2006 IEEE.

cognitive limitation by biasing their exploration towards patterns or structure that they notice. This could lead to a further difficulty if the initial projection is misleading by creating cluster memberships that bias the analyst’s investigation towards a similar solution instead of enabling any solution. Some techniques for exploratory data analysis recognize these drawbacks, attempting to compensate for them by using randomized initial displays or by providing interaction methods to learn how to cluster the entire dataset based on a few user-driven observation classifications [317].

4.3.1 Visualization Challenges

Most dimension reduction algorithm outputs are shown in scatter plots or node-link diagrams. These scatter plots come with inherent issues, such as difficulties in displaying and interpreting the dimensions that result from an MDS projection. When dealing with large datasets, the scatter plot or node-link representation of the dimension reduction output runs a high risk of overplotting, especially if the spatialization exhibits clear clustering in the lay-

out. One solution for overplotting is to abstract a cluster of observations into a single glyph to represent a collection of observations, such as suggested by the Splatterplots implementation [218]. An alternative is to filter the number of observations visible in an overdrawn region, keeping a representative ratio of each cluster in the overdrawn region [55].

While the natural representation of the dimension reduction output uses a spatial projection like a scatter plot or node-link diagram, the possibilities for representing cluster membership are much more diverse. In addition to demonstrating clusters using a collection of nodes in close spatial proximity, cluster membership can be encoded with colors or glyphs. Even then, a number of design decisions can be made for how best to express these memberships by color and shape.

Saket et al. [265] evaluate several encodings of cluster information (see Figure 4.5 for a visual representation of each of these encodings), relating each to node-based tasks (for example, “Given node X, what is its background color?”) and group-based tasks (“Given nodes X and Y, determine if they belong to the same group”). They found that the addition of group encodings does not negatively impact time and accuracy on node-based tasks. As would be expected, group-based tasks were best solved by node-link-group encodings. This outcome suggests that the visual representation used to encode the clusters in the projection depends on the tasks that the system addresses.

Jianu et al. [168] perform a similar evaluation on four visual representations, including a node-link diagram similar to that studied by Saket et al. as well as three other visual representations that are shown in Figure 4.9. Linesets [9] include link colors that match the node colors representing cluster membership, highlighting connections between nodes that are in the same cluster or group (top-right of Figure 4.9). GMap [129] is a space-filling representation that renders a geographic-like map for clusters, containing all of the nodes in a colored region similar to the node-link-graph representation studied by Saket et al. (bottom-

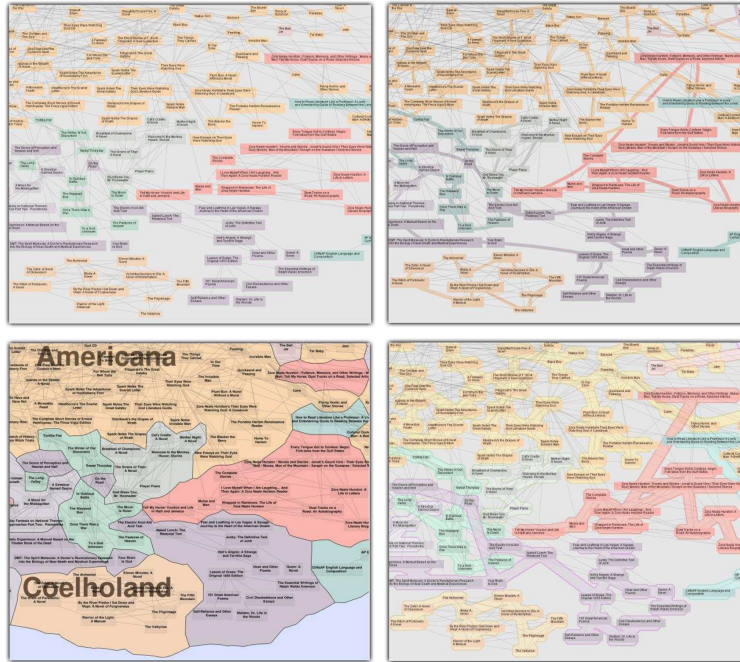


Figure 4.9: Four options for displaying cluster membership as studied by Jianu et al [168]. In addition to a node-link representation similar to that included by Saket et al., this study included Linesets [9], GMap [129], and BubbleSets [74]. © 2014 IEEE.

left of Figure 4.9). Finally, BubbleSets [74] draws isocontours around clusters, effectively balancing the Linesets and GMap representations by using the isocontours to highlight links connecting members of the same cluster but becoming space-filling in regions with high node density (bottom-right of Figure 4.9). This study found that BubbleSets was the superior representation for group-based tasks, but that encoding group information onto node-link diagrams adds a 25% time penalty onto network-based tasks, a conflict with the conclusion of Saket et al. Clearly, more research is needed in this area to resolve such conflicts.

In addition to the above, another method for visualizing clusters in a scatterplot or node-link diagram is to enclose nodes from individual clusters in a convex hull [317]. Because k -means solves for convex clusters based on a distance from an observation to the nearest cluster centroid, a convex hull visualization may be the most natural visualization representation for a k -means clustering output.

Moving away from scatter plot and node-link representations, an alternative representation for clusters is to encode topics into a streamgraph. For example, Liu et al. use streamgraphs to encode related text keywords into topical collections, using the streamgraph to show how the importance of those topics and keywords changes over time [208].

4.3.2 Algorithm Order Visualizations

Designers have an additional choice regarding which features are emphasized in the visual representation. For example, should the spatial layout of the dimension reduction be emphasized over the cluster assignments? Alternatively, should the cluster assignments inform the layout of the observations? Should we attempt to balance the two outputs? How much of an impact should the algorithm order play in the final layout? The order in which we execute the dimension reduction and clustering algorithms should have some impact on the outcome of the visualization, but the degree to which this execution order is emphasized can vary by system goals. Here, we describe potential visualization properties for each of the pipelines described in Sect. 4.2.1.

Independent Algorithms: Consider the first pipeline from Figure 4.3, in which both algorithms execute independently and in parallel. One potential outcome of this pipeline is to represent clusters using convex hulls. Here, the dimension reduction algorithm operates to find an ideal layout, while the clustering algorithm separately finds an ideal cluster set. When combining the outputs, a potential result is a cluttered visualization that is somewhat ambiguous in the cluster assignments of some observations due to intersections between the clusters. A potentially better solution, used by iVisClustering [196], is to use nodes colored by class in cases of cluster occlusion such as these. Another solution that allows the convex hulls to remain is to implement layout constraints (such as those in IPSep-CoLa [106]) so that

objects that clearly belong to different clusters are visibly separated in the spatialization. However, this requires prior knowledge of key cluster-defining objects, or an initial clustering computation that precedes the main clustering process. This also defeats the goal of the pipeline by removing the separation between dimension reduction and clustering algorithm execution.

Dimension Reduction Preprocessing for Clustering: In this pipeline, the output of the dimension reduction algorithm is fed into the clustering algorithm, enabling clustering on the low-dimensional reduced data rather than on the initial high-dimensional data. Because clusters are drawn based on the proximity of observations in the projection, it is unlikely that these clusters will intersect. As noted previously, executing the clustering algorithm on the dimension-reduced data may not produce an optimal clustering on the high-dimensional data, which could affect the analyst's comprehension of the projection.

Clustering Preprocessing for Dimension Reduction: In the reverse of this process, we now cluster in the initial high-dimensional data, and use some of that information such as the cluster assignments to inform the dimension reduction. Such a visualization will likely result in visibly separated clusters as in the previous case, though perhaps even more separated because space can be artificially added between the clusters. Again, because we execute the dimension reduction algorithm on the cluster assignment information (or other cluster algorithm output) rather than on the initial high-dimensional data, the dimension reduction projection may not be optimal and could also affect the analyst's comprehension of the projection. More clearly stated, two points that the dimension reduction algorithm judges to be somewhat similar (but not similar enough to belong to the same cluster) may have an artificially large distance applied between them in this projection.

One Algorithm Implicitly Includes the Other: A pipeline in which only one algorithm is executed to perform both the dimension reduction and clustering functions has

inherent limitations depending on which algorithm is performed. For example, if the dimension reduction algorithm is executed and clustering is applied only on the result of the dimension-reduced spatialization, the clustering will likely be far from optimal but the dimension reduction will be ideal. This could result in a visualization in which, for example, the clusters are simply assigned based on x -position in the projection.

Global and Local Algorithm Combinations: The global and local pipeline describes the dimension reduction algorithm as responsible for the global layout, while the clustering algorithm is responsible for local refinements and layout. These algorithms work together to create an overall layout in which the dimension reduction algorithm effectively lays out the clusters in a meaningful manner while the internal structure of each cluster is maintained by the clustering algorithm. As such, the fine details of the projection will not be as accurate spatially as the dimension reduction outcomes in the Independent Algorithms and Dimension Reduction Preprocessing for Clustering pipelines, and the clustering is still executing in part on the low-dimensional projection. However, the layout should be relatively clean and understandable, and the overall structure of the projection (e.g., the relative positions of the clusters) will be meaningful.

Iterative, Alternating Algorithms: The final pipeline in Figure 4.3 includes both the dimension reduction algorithm and the clustering algorithm working simultaneously and collaboratively to structure a projection that is near-optimal for both representations. As such, this structure may produce the best visualizations with respect to the meaning of the data, albeit at the cost of runtime efficiency.

A number of further design decisions can be incorporated into the visualization. We have the option to emphasize the relative distance between clusters more than the relative distance between pairs of observations. Thus, the visualization space is clusters of observations that are obviously separated from each other in the space, possibly with another iteration of

the dimension reduction algorithm performed on each individual cluster to generate a local layout. As yet another alternative, if the analyst is most interested in the clusters in the projection, the emphasis could also be placed on the distance between each observation and the centroid of the cluster that it belongs to. Clusters could also be artificially separated by a secondary execution of the dimension reduction algorithm, but the superior layout determination is dependent on the distance between each observation and a centroid.

We noted in Sect. 4.2 that it is not possible to combine all pairs of dimension reduction and clustering algorithms. Likewise, it is not possible to include all visual representations of dimension reduction and clustering in the same visualization. For example, dendrograms are often used to show hierarchical clustering; however, dendrograms are not a useful visual encoding for dimension reduction algorithms.

4.4 Discussion

Combining dimension reduction and clustering algorithms into the same visualization system provides a number of opportunities for visualization and interaction design. A system in which the two algorithm classes cooperate for exploratory data analysis results in a relationship in which the projection space (the outcome of the dimension reduction algorithm) helps to explain the meaning of the clusters in the space, while the clusters themselves help to explain the meaning of the space.

Including a machine learning aspect into a visualization system to permit the dimension reduction and clustering algorithms to learn from the actions of the analyst presents a number of additional challenges for interaction design. In particular, the overloaded space metaphor discussed in Sect. 4.1.1 causes challenges, as interactions within the system must be mapped to at least one algorithm and may ambiguously be mapped to both. For example, if an

analyst drags and drops a datapoint to reposition it in space, but the new coordinates did not result in a cluster reassignment, should the clustering algorithm learn nothing, or did the analyst provide some “fuzzy” clustering feedback to the algorithm? A notion of iterative refinement, in which the analyst gradually trains the algorithms and offers corrections to mistakes at each iteration is necessary in these cases. Such an iterative refinement process mimics Pirolli and Card’s Sensemaking Process [248].

Maintaining an analyst’s mental map during layout adjustments is a well-studied problem [224], and is another factor that should be considered in visualization and interaction design for dimension reduction and clustering systems. ForceSPIRE and Andromeda approach this mental map challenge in different ways. ForceSPIRE, using a force-directed layout, maintains the positions of nearly all observations during an interaction, only altering the positions of observations near the interaction [115]. In Andromeda, on the other hand, it is possible that all observations could move the entire distance across the space. The system cognitively aids the analyst to understand such broad changes with an animation slider, affording the analyst with the ability to incrementally follow the post-interaction transition, as well as a layout stabilization module to suppress the rotation invariant property of the Weighted Multidimensional Scaling dimension reduction algorithm [273].

This work focuses on exploring the breadth of design options available to visualization researchers when combining dimension reduction and clustering algorithms. Our goal with this work is to highlight many of the decisions that exist in this design space, spurring further exploration of this space with new tools. While we present a number of design questions that must be addressed in creating such a visualization system, we do not claim to answer any of these questions, as the answers to many of them depend on the tasks and goals of the system.

Table 4.2: A summary of the design challenges and questions discussed throughout the chapter regarding the combination of dimension reduction and clustering algorithms.

Section	Design Decision
4.2	What properties of the data is the visualization seeking to highlight? Which properties of the data are the system and analyst trying to discover? Should the primary goal of the visualization system be emphasizing observation relationships, clusters of observations, or both? Should the dimension reduction and clustering algorithms use the same distance function (if possible), or should each algorithm use an independent similarity method?
4.2.1	Which order and interaction of dimension reduction and clustering algorithms best models the task that the visualization system is addressing?
4.3.1	How can we encode distances and cluster membership information when both algorithms are present?
4.3.2	As the dimension reduction and clustering algorithms are competing in the same visualization, what features should be emphasized in the visualization to best address the problem?

4.5 Conclusion

The combination of dimension reduction and clustering algorithms represents an immense design space, including considerations of algorithm selection and order, tasks, and visualization. In this chapter, we have provided a survey of each of these considerations, describing existing research and discussing relevant design decisions applicable to current and future systems (summarized in Table 4.2).

Returning to our discussion of the “Be the Data” interaction first addressed in Sect 4.1, we saw a smooth transition from discrete to continuous thinking. The students initially formed the clusters of Edible and Inedible animals and then positioned those clusters in space, initially mimicking the cluster preprocessing pipeline. The transition from this projection into a spectrum of Edibility amounts to iterative and interactive refinement of those initial clusters into a broader projection. Without the interaction component, the pipelines could not successfully model this student behavior.

An additional component of this design space that has not been addressed in this chapter is methods for interacting with these visualization systems. We begin discussing this space in the next chapter.

Chapter 5

Dimension Reduction and Clustering Interactions

“With respect to what” was first described as a usability issue with interactive projections by Self et al [273]. In the Andromeda system, analysts are presented with the two-dimensional output of a WMDS dimension reduction computation. By performing direct manipulation interactions on the observations in the projection, analysts communicate desired similarity relationships to the system. This triggers a learning routine that attempts to create such relationships in the projection by altering the weights applied to the dimensions.

The usability issue that emerges from this interaction technique revolves around interpreting the analyst’s intent appropriately. That is, when the analyst moves an observation to a new position, what is that movement in relation to? Is this relationship assumed or somehow explicitly communicated by the analyst? Possible interpretations for repositioning an observation in the projection include but are not limited to moving the observation away from the source, moving the observation towards a target, and moving the observation with respect to some other observation(s) within the projection. In other words, what did the analyst move, and *with respect to what?*

Resolving this “with respect to what” problem is increasingly important in order to capture the intent of the analyst. Introducing clustering algorithms as a further complexity can help to resolve this challenge in part (described in the next chapter), as implicit clusters

often form naturally in dimension-reduced projections that display similarity relationships. Defining these clusters explicitly also enables explicit relationship communication to the system.

However, ambiguity in the interpretation of these interactions does still exist after explicit clustering has been introduced, as described in our motivating example in Section 5.2. In this work, our goal is to more thoroughly detail the interaction space for the simultaneous use of dimension reduction and clustering algorithms, particularly in interactive projections that feature a learning component. Specifically, we claim the following contributions:

1. An overview of the “with respect to what” problem space and survey of existing solutions to the problem.
2. A discussion of the implications and complexities of the “with respect to what” problems as it related to interactive projections that depict both observations and their clusters.
3. An exploration of the interaction space with respect to dimension reduction and clustering algorithms in the same interactive projection.

5.1 Background

After displaying a visualization of dimension-reduced and clustered data, the next step is to provide interactions to afford user exploration through the dataset. Many studies have been performed and taxonomies generated for interacting with high-dimensional data in a data analytics context [11, 45, 310, 334].

In the context of exploring dimension-reduced data projections, two primary methods exist for modifying an underlying distance function: Parametric Interaction and Observation-Level

Table 5.1: Sample interactions, organized by type of interaction (rows) and by the type of algorithm affected by the interaction (columns).

	Dimension Reduction	Both	Clustering
PI	Rotate the projection	Modify the weight on a dimension Select a different distance function	Modify the max/min radius of a cluster Change the number of clusters sought
OLI	Reposition an observation external to clusters or within a single cluster	Reposition an observation into a different cluster	Change cluster membership Merge several clusters or split a cluster
Surface	Measure a distance between observations	Details-on-demand to obtain attribute values	Count the size of a cluster Annotate a cluster

Interaction [272]. Surface-level interactions are also often incorporated into visualization systems, though these do not modify the underlying model. We begin this section by discussing these interaction techniques and some representative tools, as well as discussing interaction techniques that address clustering challenges. We follow this with a discussion of potential interaction techniques that can support interaction with both dimension reduction and clustering algorithms simultaneously.

5.1.1 Current Interaction Techniques

Parametric Interaction (PI) refers to manipulating parameters directly in order to create a new projection and/or clustering assignment. This presents a difficulty to novice or non-mathematically-inclined analysts, who may not understand how to update a set of weights to create the dimension-reduced projection that they desire. In contrast, Observation-Level Interaction (OLI) refers to direct manipulation of the observations, which in turn triggers a back-solving routine to learn new parameters [109, 157, 199]. In this way, OLI hides the manipulation of the model from the analyst, allowing the analyst to perform more natural direct manipulation interactions with the observations themselves. In Andromeda [273], PI allows analysts to modify weights on the dimensions to modify the distance function directly by interacting with sliders, while OLI uses an inverse MDS computation to interpret the semantic meaning of the interaction in order to solve for those weights.

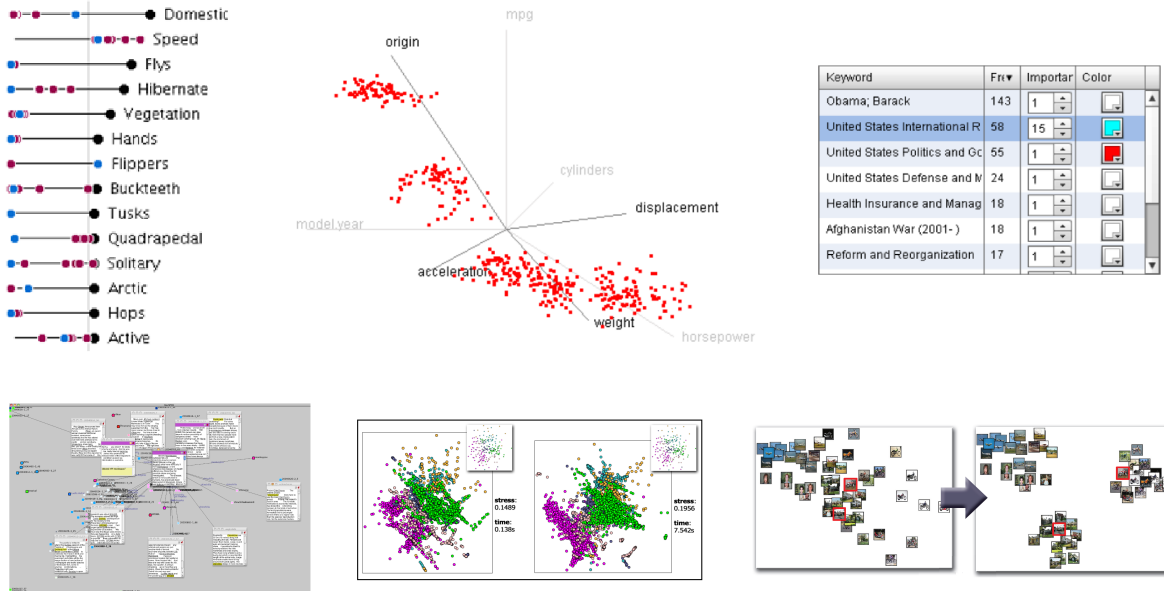


Figure 5.1: A selection of interfaces and tools that support Parametric Interaction or Observation-Level Interaction. The upper row shows PI interfaces that include slider bars from Andromeda (PI view) [273], Star Coordinates [174], and SpinBox widgets from STREAMIT [10]. The lower row shows OLI interfaces from StarSPIRE [38], Paulovich et al. [239], and Mamani et al. [214]. Included under Fair Use, 2019.

The upper row of Figure 5.1 shows sample examples of PI from recent visualization systems, complemented by some representative interactions in the upper row of Table 5.1. Horizontal and vertical slider bars are frequently utilized to enable analysts to interact with model parameters, despite the fact that these model parameters have a variety of contexts. Some of these sliders, such as those in Andromeda [273], include additional glyphs on the sliders to show the values of selected observations on each dimension. In addition to slider bars, other techniques have been utilized to support the manipulation of model parameters, such as the SpinBox widgets of STREAMIT [10] and the transforming axes of Star Coordinates [174]. PI techniques can also be extended to interact with dimensions as well as observations, as shown by Turkay et al [303].

As seen in the lower row of Figure 5.1 and discussed in Sect. 4.3, scatter plots and node-link diagrams are the overwhelming favorite for displaying dimension-reduced projections, includ-

ing those that support OLI. Despite the ubiquity of these visual representations, individual OLI systems display unique features and properties, such as supplementing the scatter plot with additional views for context [42], supporting PI in addition to OLI [273], including local transformations[214], and focusing exclusively on textual data [38].

An additional consideration for OLI is the “With Respect to What” problem detailed by Self et al. [273], which is the fundamental challenge of using rigid algorithms to interpret the ambiguous meaning of an interaction that involves dragging a node from one part of the display to another. Andromeda solves this challenge by defining a radius at both the starting and ending point of the interaction, implying that the analyst is moving an observation away from all other observations within x pixels of the source and towards all other observations within x pixels of the destination of the interaction, though the analyst is afforded the ability to deselect observations that do not apply to the interaction [273]. Points contained within this radius are highlighted in the visual representation, allowing analysts to clearly see the interaction targets that they are expressing within the projection [158].

In addition to Parametric and Observation-Level Interactions, the introduction of clusters affords a variety of cluster-based interactions that can support sensemaking. To begin, OLI can be applied to clusters, including such interactions as moving clusters together and further apart to reflect similarities and differences between clusters, as well as transferring that information either to the weights on the clusters or the weights on the nodes. We can also apply parameter tuning to clusters at a global level, changing the number of clusters or the radius of all clusters, or we can tune the parameters of individual clusters, creating a collection of clusters with a variety of radii. The Vizster system, for example, includes a PI-style slider bar to change the number of clusters displayed in the X-ray view [148].

Clusters also introduce new cluster-specific interactions, such as cluster merging, splitting, and creation [63, 150], cluster annotation [175], and hierarchies of clusters [228]. Performing

any of these interactions can communicate semantic information back to the system, re-executing the pipeline that may or may not also include re-executing the dimension reduction algorithm as a result of this user interaction.

5.1.2 Combined Interaction Techniques

The pipelines discussed in Sect. 4.2.1 naturally support the Parametric, Observation-Level, surface-level, and clustering interactions discussed in the previous subsection. Interactions in general can be designed for each of these pipelines individually, but it is also useful to consider interactions that can have meaning to both the dimension reduction algorithm and the clustering algorithm simultaneously. To do so means facing similar ambiguity that is addressed by the “With Respect to What” problem and the issue of overloaded space.

For example, consider an analyst who is interacting with the clustering assignment in a projection. Regardless of whether the analyst is interacting with high-dimensional or low-dimensional clusters, dragging an observation from one cluster to another is a natural interaction to correct a misclassification. However, the cause of that misclassification may be unknown to the analyst. Perhaps the analyst is interacting with a system that implements the dimension reduction preprocessing pipeline. If that is the case, then the analyst may be correcting a misclassification that results from the clustering operating on the projected low-dimensional data. Thus, the goal of the system should be to learn from that interaction, with the goal of getting closer to the ideal high-dimensional clustering.

Alternatively, if the analyst is interacting with a system that implements clustering on the high-dimensional data, then performing the same interaction is correcting for a case where the heuristic clustering algorithm did not find the optimal solution. The system can still learn from this interaction to correct future clusterings, but the different cause of the misclas-

sification should result in a different model update. These two misclassification corrections may be semantically identical to the analyst who seeks to correct an error, but the underlying mechanics that caused and must correct the misclassification are different.

The same is true of an analyst interacting with observations in a dimension-reduced projection. If an analyst drags an observation, it may simply be that the analyst wishes to adjust the strength of the relationship between two observations. However, adjusting the strength of a relationship calculated on the high-dimensional data is inherently different than adjusting the strength of a relationship calculated on cluster algorithm output. And does the semantic meaning of the interaction change if that drag interaction crosses a cluster boundary?

The introduction of explicitly-defined clusters allows for a formal target against which to judge interactions. When explicit clusters are defined, the analyst has four high-level “with respect to what” operations: (1) moving an observation into a cluster, (2) moving an observation out of a cluster, (3) moving an observation from one cluster into another, and (4) moving an observation without changing cluster membership [317]. Each interaction can be designed to have an effect on both the dimension reduction algorithm and the clustering algorithm. Keeping an observation within a cluster, or dragging it from one cluster into another, provides information to the clustering algorithm that the classification is either correct or incorrect. At the same time, relocating an observation to a different position communicates suggested distance information between the moved observation and one or more additional observations in the projection. Each of these algorithms can thus work to update the weight vector, leading to a projection and clustering update with this new information.

When mapping interactions to the pipelines summarized in Figure 4.3, choosing the primary target of the interaction is important, even when an interaction affects both algorithms. In the previous example, the pipeline is implemented with the interaction primarily occurring on the clusters, changing the cluster assignment of observations in order to update the dimension

reduction projection [317]. In contrast, “Be the Data” also implements the same pipeline but with an interaction primarily on the observation layout, using the dimension reduction algorithm to update the clusters [57]. These systems are both implementations of the same pipeline, but place the interaction on different algorithms to answer different questions about the high-dimensional data. Thus, interactions can be considered independent of the pipelines.

A further open question to be addressed regards interactions on the clusters themselves. If an analyst drags a cluster or interacts with it in another manner, what adjustments should be made to the observations and relationships within that cluster, as well as the relationships that cross that cluster boundary?

5.1.3 “With Respect to What”

These interaction schemes can be applied in a wide variety of applications. Each method has implications for how the “with respect to what” problem can or should be solved. For example, using control points within a visualization is a common method for enabling interactive and iterative refinement of the projection [38, 84, 93, 114, 169, 214, 225, 239, 276, 319]. Control points often take the form of analyst-selected and manipulated points within the projection, but these control points can also be represented as anchors on the projection boundaries as well. In either case, the analyst is manipulating a given point with respect to the entire visualization. That is, this interaction is meant to have an effect on a global scale rather than performing local refinements. This concept is reflected in the fact that a single movement of one control point typically results in all other non-control points moving themselves in relation to the control point’s new location.

Rather than use control points, some tools instead use manipulated points to describe desired pairwise distances in a projection [42, 98, 262, 272, 273, 274, 317]. As a result, the informa-

tion that is communicated through this interaction is a desired set of distances expressed as relative pairwise distances between the moved points, which often reflect similarity/dissimilarity relationships in the data. Using these relative pairwise distances, the system *learns* a new distance metric, typically by updating the parameters of the chosen distance function. The new distance function is then applied to all projected data, not just the interacted points, to produce an updated visualization. The implied “with respect to what” in such interactions is limited to the points the analyst interacted with; all other points are ignored until the data is reprojected.

There are still other types of interactions which address this “with respect to what” problem. Podium [313] allows analysts to interactively alter the rank of any item in a table. Podium explicitly defines this interaction to be with respect to the other rows that changed ranks as a result of the interaction. Additionally, ReGroup [13] enables interactive cluster formations in which each additional item that is added to a cluster results in updating a list of suggested items to add to the cluster. Thus, this interaction is with respect to the existing items within the cluster. Andrienko et al. take yet another approach in which cluster definitions can be interactively altered by the analyst, such as merging or splitting clusters [16]. Such interactions are with respect to the involved clusters (i.e., the clusters that are being merged together or which cluster is being split). These examples demonstrate the variety of manners in which the “with respect to what” problem can be addressed, indicating the vast design space present in this area.

5.1.4 Pipelines in Visual Analytics

Pipeline representations are often used to visually communicate the flow of data within a visualization system from one processing component to the next. The convention of showing

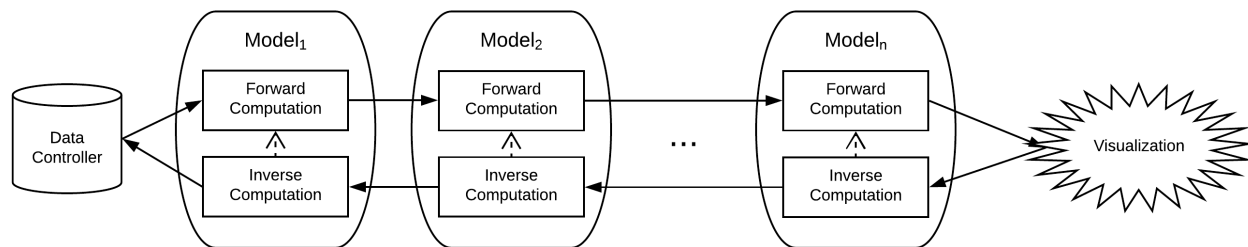


Figure 5.2: The bidirectional, multi-model pipeline for semantic interaction proposed by Dowling et al [97]. Included under Fair Use, 2019.

the generation of a visualization by data flow to the right and interaction handling via flow to the left dates to at least the information visualization pipeline [49]. With respect to visual analytics and interactive projections, Endert et al. provide a generic pipeline representation of a model-driven system [113], which was further expanded into a multi-model system by Bradel et al [38].

More recently, Dowling et al. propose a bidirectional, multi-model pipeline for semantic interaction applications [97]. We reproduce this pipeline in Figure 5.2. In this pipeline structure, the projection is created by processing a sequence of “Forward Computations,” while the interactions are handled by processing a similar but reverse sequence of “Inverse Computations.” In this chapter, we adopt a similar pipeline approach to summarize models and their interactions, though we use the terminology “Projection Computation” and “Interaction Computation” rather than “Forward” and “Inverse.”

In the case of a visualization system that incorporates dimension reduction and clustering algorithms into the same interface, each of these algorithms would represent a separate model in the sequence. The order of these models in the sequence can therefore change both the meaning and the behavior of the visualization. Running the dimension reduction computation before the clustering computation implies that a dataset is reduced from the high-dimensional space to the low-dimensional space, after which the clustering algorithm is performed on the low-dimensional data.

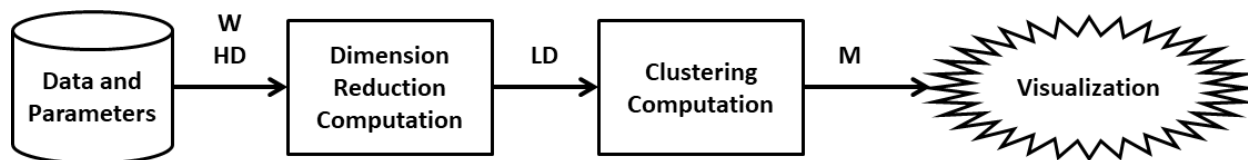


Figure 5.3: The “Dimension Reduction Preprocessing for Clustering” projection pipeline from the previous chapter, annotated with the structure of input and output data. In this and many future figures, W =dimension weights, HD =high-dimensional data, LD =low-dimensional data, M =cluster membership.

In contrast, running the clustering computation before the dimension reduction computation implies that the clustering is being performed on the high-dimensional space, after which the dimension reduction is performed perhaps on the cluster centroids. The previous chapter presents a broad discussion of these and other projection pipelines. As our discussion of these projection and interaction pipelines will go into further depth, we further annotate these pipelines with information about the structure of the input and output data, as seen in Figure 5.3.

5.2 Motivating Example

To motivate our discussion of the dimension reduction and clustering interaction space, consider the example shown in Figure 5.4. In this example, an analyst is provided with a dimension-reduced projection of an animal dataset [195], positioned according to their attribute relationships with initially equal weights. A clustering algorithm then groups the observations into discrete categories, following the “Dimension Reduction Preprocessing for Clustering” pipeline described in the previous chapter. After viewing the projection, the analyst chooses to reposition the Grizzly Bear observation, removing it from one cluster and placing it into another. With this simple interaction, the analyst could be trying to convey and number of possible intents to the system.

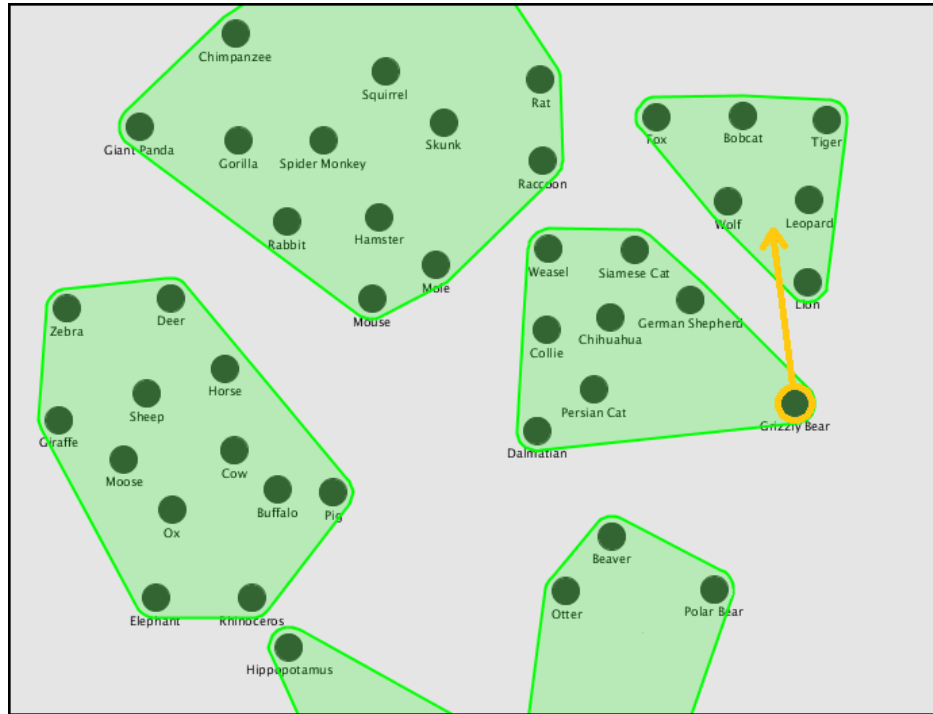


Figure 5.4: An analyst repositions the Grizzly Bear observation within the projection, indicated by the orange arrow.

Perhaps the analyst is looking specifically at the relationships between the animals in the projection. For example, the analyst could be trying to convey a relationship about the starting position of the observation (“the Grizzly Bear is not similar to the other animals near the source”) or a relationship about the ending position of the observation (“the Grizzly Bear is more similar to the other animals near the destination”). There is also the question of how many observations the analyst considers; the analyst could be trying to communicate a relationship with respect to the closest observation (“the Grizzly Bear is most similar to the Lion”), the closest n observations, all observations in a cluster, or all observations in the projection. These types of relationships would be best handled by a Dimension Reduction Model.

Alternatively, the analyst may have mapped some semantic meaning onto the cluster groupings in the projection by trying to communicate a membership assignment based on those

groups (“the Grizzly Bear is a better fit in the Predators cluster than in the Pets cluster”). Such relationships could incorporate both the source and the target cluster, or perhaps a case where the target is irrelevant (“the Grizzly Bear appears to be an outlier in the Pets cluster and belongs elsewhere”) or the source is irrelevant. These relationships would be best handled by the clustering algorithm.

The analyst may also be trying to communicate a relationship that includes both observations and clusters. In such cases, the relationship may be relevant to all observations within the cluster (“the Grizzly Bear is more similar to the observations in the target cluster than the source cluster”), or the precise positioning of the observation within the cluster may be important (“the Grizzly Bear belongs in the Predator cluster, but it is not similar to the small predator (Fox)”).

The examples in the preceding paragraphs suggest two primary dimensions to consider when judging the intent of the interaction. First, the interaction could be applied to the observations, the clusters, or both. Second, the interaction could be applied to a variety of cardinalities: the nearest observation, the nearest n observations, all observations within a cluster, or all observations in the projection. These dimensions are summarized with respect to the Grizzly Bear observation and Predators cluster in Table 5.2 and are expanded upon in the following sections. In addition to these two dimensions, it is also worth considering further details such as whether the important component of the interaction is at the source location, the target location, or both. Finally, it is worth noting whether the analyst is thinking in a high-dimensional space or a low-dimensional space when performing the interaction.

Table 5.2: A collection of example intents and interactions that an analyst could communicate via repositioning an observation or a cluster in a projection.

		Analyst Repositions		
		Observation	Cluster	
With Respect to What	Nearest 1	Observation	Intent: The Grizzly Bear is most similar to the Polar Bear. Interaction: Drag the Grizzly Bear close to the Polar Bear.	Intent: The Predators cluster is dissimilar from the Blue Whale observation. Interaction: Drag the Predators cluster away from the Blue Whale observation.
		Cluster	Intent: The Grizzly Bear is similar to other members of the Predators cluster. Interaction: Drag the Grizzly Bear into the Predators cluster.	Intent: The Predators cluster is dissimilar from the Large Herbivores cluster. Interaction: Drag the Predators cluster away from the Large Herbivores cluster.
	Nearest n	Observations	Intent: The Grizzly Bear is similar to other carnivorous animals. Interaction: Move the Grizzly Bear closer to the Lion, Tiger, and others nearby.	Intent: The Predators cluster is dissimilar from the aquatic animals. Interaction: Drag the Predators cluster away from the seal, walrus, and other nearby animals.
		Clusters	Intent: The Grizzly Bear is a predatory animal. Interaction: Move the Grizzly Bear closer to the clusters that contain predatory animals.	Intent: The Scavenging Predators cluster is similar to the small actively hunting and large actively hunting predators. Interaction: Move the scavenging predators cluster closer to the other Predators clusters.
	Cluster	Single	Intent: The Grizzly Bear belongs in the Scavenging Predators cluster. Interaction: Move the Grizzly Bear within the Predator cluster boundary.	Intent: The Predators cluster are similar to the Grizzly Bear observation. Interaction: Drag the Predators cluster closer to the Grizzly Bear.
		Multiple	Intent: The Grizzly Bear is a predator. Interaction: Drag the Grizzly Bear somewhere among the Small Predators, Large Predators, and Scavenging Predators clusters.	Intent: The Scavenging Predators cluster is a subset of the Predators. Interaction: Move the scavenging predators cluster within the Predator cluster boundary.
All of the	Observations	Intent: The Grizzly Bear is more similar to the predatory animals on the left than the herbivorous animals on the right. Interaction: Drag the Grizzly Bear from the left side of the projection to the right.	Intent: The Scavenging Predators cluster is more similar to the other carnivorous animals on the left than the herbivorous animals on the right. Interaction: Drag the Scavenging Predators cluster from the left side of the projection to the right.	
	Clusters	Intent: The Grizzly Bear is more similar to the predatory animal clusters on the left than the herbivorous animal clusters on the right. Interaction: Drag the Grizzly Bear from the left side of the projection to the right.	Intent: The Scavenging Predators cluster is more similar to the carnivorous animal clusters on the left than the herbivorous animal clusters on the right. Interaction: Drag the Scavenging Predators cluster from the left side of the projection to the right.	

5.3 Interactions on Observations

In this section, we consider the possible interpretations that result when an analyst repositions an observation in the projection. We begin by describing observation interactions that affect observations, before moving into interactions that affect clusters, and conclude with

interactions that affect both. In this analysis, we consider the pipeline representations most natural to each interaction. We summarize the properties of these interactions in Table 5.3.

5.3.1 With Respect to Other Observations

As detailed by the left column of the intents and interactions in Table 5.2 and summarized in the previous section, when an analyst repositions an observation, the system must determine what the analyst is moving the observation with respect to. The analyst might be repositioning the observation to move it away from something near the source, towards something near the target, or relative to any other observation in the projection. The analyst might also be repositioning the observation relative to the position of just a single observation or a collection of n observations.

As the Dimension Reduction Model is responsible for the positioning (and repositioning) of observations, it is most natural to use the Dimension Reduction Model to interpret the intent of the analyst in each of these cases. The projection computation of the Dimension Reduction Model takes dimension weights and high-dimensional data as input and generates low-dimensional data as output. In contrast, the interaction computation of this model should consider relative positions of interacted observations in the low-dimensional space, and will generate new weights that will cause such relationships to appear in a subsequent projection when applied to the immutable high-dimensional data. Such an approach is used by Andromeda [272, 273, 274], as described at the beginning of this chapter. These relationships are summarized in the following equations, and are shown graphically in Figure 5.5.

$$\begin{aligned}
 LDdata &= \text{DR_PROJECT} (weights, HDdata) \\
 weights' &= \text{DR_INTERACT} (HDdata, LDdata')
 \end{aligned}$$

Table 5.3: A summary of the observation interactions discussed in this paper by cardinality, the importance of the interaction source, target, or both, and whether an analyst is thinking high-dimensionally, low-dimensionally, or both.

	Cardinality	Source/Target	High-D/Low-D
Observation–Observation Similarity			
Move observation towards another observation	1:1	T	LD
Move observation away from another observation	1:1	S	LD
Move observation towards several observations	1: n	T	LD
Move observation away from several observations	1: n	S	LD
Observation–Cluster Similarity			
Move observation towards a cluster	1:1	T	B
Move observation away from a cluster	1:1	S	B
Move observation towards several clusters	1: n	T	B
Move observation away from several clusters	1: n	S	B
Cluster–Observation Similarity			
Move cluster towards an observation	1:1	T	B
Move cluster away from an observation	1:1	S	B
Move cluster towards several observations	1: n	T	B
Move cluster away from several observations	1: n	S	B
Cluster–Cluster Similarity			
Move cluster towards another cluster	1:1	T	B
Move cluster away from another cluster	1:1	S	B
Move cluster towards several clusters	1: n	T	B
Move cluster away from several clusters	1: n	S	B
Observation Change in Membership			
Move observation into cluster	1:1	T	HD
Move observation out of cluster	1:1	S	HD
Move observation between clusters	1: n	B	HD
Move observation external to clusters	1: n	B	HD
Move observation within a cluster	1:1	B	HD

Determining the other component(s) involved in the interaction is an additional challenge, particularly when differentiating between movements with respect to 1, n , or all observations. The most straightforward solution to this challenge is to provide analysts with a selection mechanism or set of mechanisms. For example, Andromeda implements a number of methods to permit the analyst to choose all elements necessary for the interaction. At the source, the nearest neighbor to the observation interacted upon is selected by default, as are all

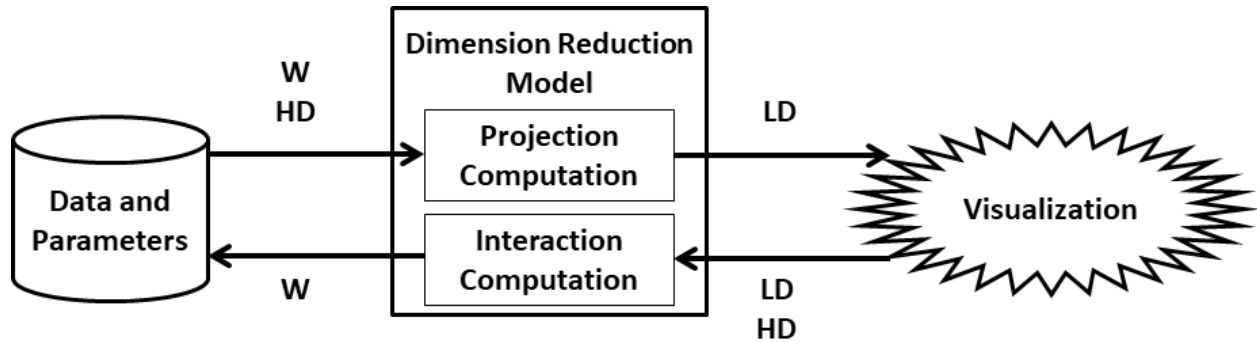


Figure 5.5: A representation of data flow using a dimension reduction model to learn a new projection. The clustering algorithm is not necessary for observation-to-observation interactions, and could be positioned either to the left or right of the dimension reduction algorithm.

observations within a set radius of the target position of the interaction. Following these default selections, the analyst is permitted to select or deselect any other observation in the projection. An example of each of these is shown in Figure 5.6, in which an analyst has repositioned the Beaver observation closer to the whales, possibly signifying an interest in exploring aquatic-dwelling animal behavior. As the nearest neighbor to the source, the Polar Bear was automatically selected as part of the interaction, as were the Blue Whale and Humpback Whale in the target radius. After this observation was repositioned, the Wolf was also selected to denote dissimilarity between land-dwelling and water-dwelling animals.

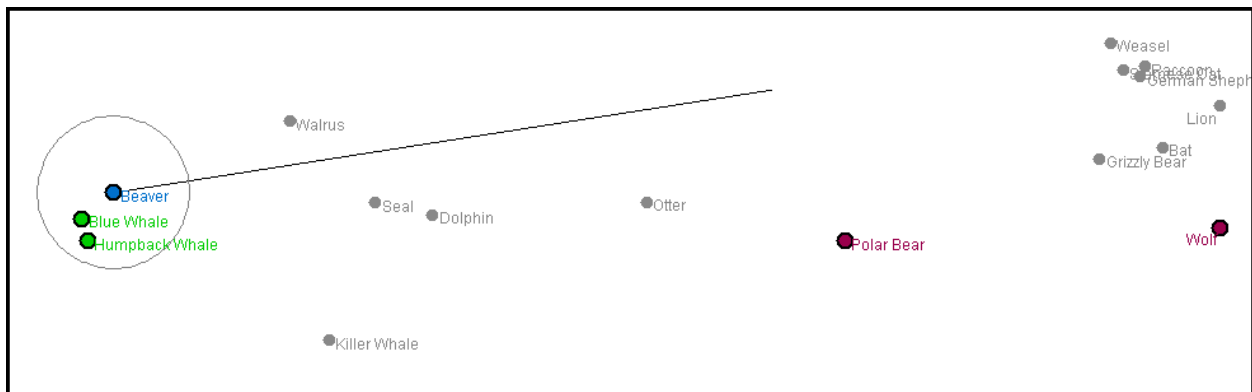


Figure 5.6: Selection interactions in Andromeda [274]: nearest neighbor selection at the source, radius selection at the target, and additional observation selection in other regions.

The further possibility exists that the analyst does not wish to alter any underlying models with the interaction they provide. Instead, they may be merely exploring the current projection. Endert et al. define these categories of exploration as exploratory and expressive: *exploratory* interactions provide an analyst with insight into the structure of the data, whereas *expressive* interactions communicate an intent to the system and effect underlying models [109]. For example, the tool introduced in the next chapter treats interactions that do not cross cluster boundaries as exploratory, allowing analysts to investigate relationships between observations within clusters without affecting the underlying learning system [317]. This is generally true for drag interactions in other systems that incorporate force-directed layouts, such as ForceSPIRE [114] and StarSPIRE [38].

That said, StarSPIRE allows for explicit interactions by having the analyst overlap the boundaries of two documents. In this case, the system interprets this interaction as the analyst expressing not just document similarity, but their immediate, close relatedness. Thus, StarSPIRE uses this interaction to increase the weight associated with all shares entities between the two documents, resulting in an updated projection that includes new documents discovered through semantic interaction foraging [38, 319].

5.3.2 With Respect to Clusters

Repositioning an observation with respect to a cluster leads to a further set of challenges, primarily centered around the means by which cluster information is encoded in the projection. This is due to the fact that the visual encoding of clusters leads to different affordances for interaction. In this subsection, we first consider clusters defined by an explicit border, as in the motivating example from Section 5.2 and Figure 5.4. After this discussion, we summarize these interactions with respect to color and cluster hierarchies.

Table 5.4: A collection of example intents and interactions that an analyst could communicate via reclassifying an observation with respect to a cluster in a projection that uses cluster boundaries.

	Analyst Reclassifies an Observation
Into a cluster	Intent: The Grizzly Bear is a large predator. Interaction: Move the Grizzly Bear observation within the Large Predators cluster boundary.
Out of a cluster	Intent: A Grizzly Bear is not a pet. Interaction: Move the Grizzly Bear out of the Pets cluster boundary.
Between clusters	Intent: The Grizzly Bear is better classified as a hunting predator than a scavenging predator. Interaction: Move the Grizzly Bear out of the Scavenging Predator cluster boundary and into the Hunting Predator cluster boundary.
External to all clusters	Intent: The Grizzly Bear is more like the large cats than the wolves. Interaction: Move the Grizzly Bear from near the Large Cats cluster to near the Wolves cluster without reclassifying.
Internal to a cluster	Intent: The Grizzly Bear is more like the large predators than the small predators. Interaction: Move the Grizzly Bear within the Predators cluster from near the small animals to near the large animals without reclassifying.

Boundaries: Explicit cluster boundaries in a projection suggest to an analyst that repositioning an observation into or out of a cluster is communicating a membership assignment to the system. Such an interaction then could be interpreted in a variety of ways: an observation is moving into a cluster, out of a cluster, between clusters, separate from all clusters, or internal to a cluster. A collection of example observation reclassification interactions and their related intents are included in Table 5.4.

Using such visual encoding of the clusters means that the Clustering Model is now necessary to interpret these interactions. Therefore, the model’s location in the system pipeline is relevant to both the projection and the interaction. As discussed in the previous chapter, running the Dimension Reduction Model before the Clustering Model in the projection direction shows clusters in the low-dimensional space. If the projection displays low-dimensional clusters, then a cluster reassignment can be interpreted as informing the system that the high-dimensional interpretation of groups in the data does not match the low-dimensional classification. In such a pipeline, the analyst is reasoning in high-dimensional space, and thus the high-dimensional data should be taken into account when interpreting the interaction.

Still, a distance between the repositioned observation and each of the source and target clusters is necessary to understand the high-dimensional relationship between these entities. Such a pipeline is provided in Figure 5.7 and is demonstrated in the next chapter, with expressive interactions that cross cluster boundaries. As with the previous pipeline in Figure 5.5, this pipeline has both a projection direction to generate the desired visualization and an interaction direction to interpret analyst intent. The new feature is the incorporation of a Clustering Model to join the Dimension Reduction Model.

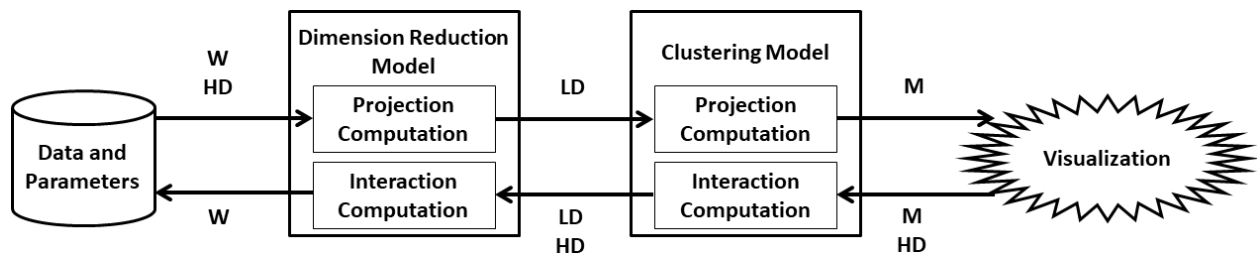


Figure 5.7: A representation of data flow using a Clustering Model to interpret a change in cluster membership, followed by learning distances with a Dimension Reduction Model, to learn a new projection.

In contrast, a pipeline that follows the “Clustering Preprocessing for Dimension Reduction” pattern implies a layout of cluster centroids of any dimensionality. In such a projection, repositioning an observation from one cluster to another implies that the high-dimensional classification of the observation does not meet the expectation of the analyst during their current exploration. As such, the interaction computation of the Clustering Model is needed (rather than of the Dimension Reduction Model) to resolve the change in cluster membership. Such a pipeline is provided in Figure 5.8.

Color: If clusters are encoded by a mechanism other than boundaries, such as color, then the natural interactions afforded by the system will change. Color is often used to demonstrate cluster assignments in systems where items belonging to different clusters may be positioned nearby in a projection [9, 63, 181]. In other words, explicit cluster boundaries

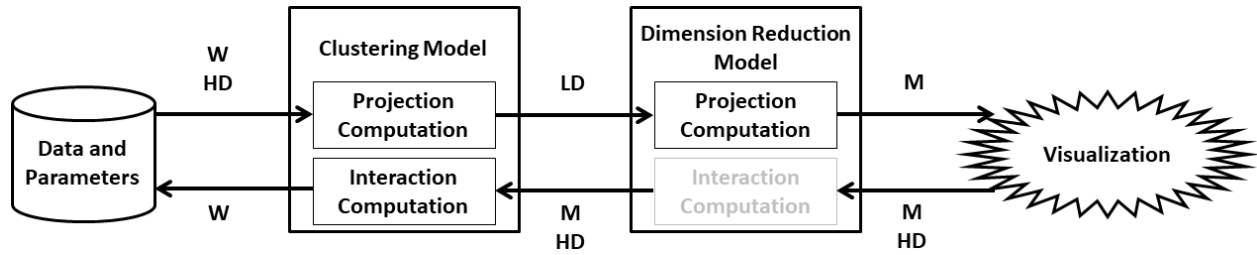


Figure 5.8: A representation of using a Clustering Model alone to interpret a change in cluster membership.

are more easily interpreted when cluster regions can be easily and accurately expressed by non-overlapping regions, implying that convex hulls are preferred in these scenarios. As such, simply repositioning an observation into a grouping of observations will not always be sufficient to communicate a new cluster assignment.

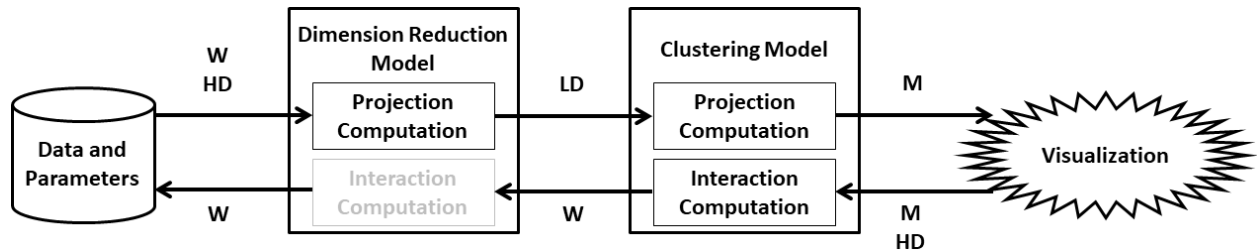


Figure 5.9: An alternative representation of data flow using a Clustering Model to interpret a change in cluster membership to learn a new projection.

Instead, an alternative cluster reassignment mechanism may be necessary. Perhaps clicking on an observation will cycle through its possible colors and therefore its cluster assignment. In such a case, the Clustering Model is solely responsible for learning from the interaction. The position of the dimension reduction algorithm is therefore dependent upon the method of projection, either identical to that shown in Figure 5.8 if the Clustering Model is first to execute, or as shown in Figure 5.9 if the Dimension Reduction Model is first to execute.

Cluster Hierarchies: If clustering is hierarchical, then this learning process becomes more complex. A system would need to evaluate where in the overall hierarchy an observation began and where it ended. Therefore, the learning relationships also may depend upon not

only the source and target cluster of the interaction, but also the parent clusters and their properties at each endpoint. As such, a recursive computation of cluster properties and weights is necessary to consider the full hierarchical structure.

With Respect to Both Clusters and Observations: In addition to the prior examples, it is possible that an analyst is communicating both position and membership information simultaneously via an interaction. Again, consider the interaction in the motivating example from Section 5.2. The analyst may wish to communicate that the Grizzly Bear belongs in the Predators cluster, while simultaneously communicating that the Grizzly Bear is more similar to the large predators in the cluster than it is to the small predators. In such a case, both the Dimension Reduction and Clustering Models are required to interpret the interaction. Again, the precise pipeline to handle such an interaction depends upon the projection meaning. If the dimension reduction output is used to inform low-dimension clustering assignments, then the overall pipeline could remain similar to that in Figure 5.7. Alternatively, if the clustering output is used to position cluster centroids via dimension reduction, then a similar pipeline could be utilized but with the model order swapped, as in Figure 5.10.

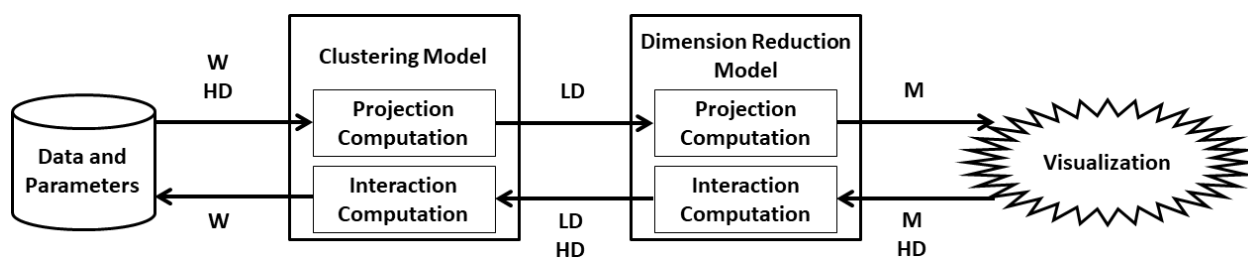


Figure 5.10: A representation of data flow using a Clustering Model first to interpret a change in cluster membership to learn a new projection.

However, there may be a need to provide additional communication between the models. In other words, learning optimal weights for the new projection could depend upon the interaction computations of the dimension reduction and clustering algorithms working together to determine the optimal configuration. This balances an updated clustering reassignment

with the precise low-dimensional coordinates of the interaction target. As such, a pipeline such as that in Figure 5.11 would be used, in which the two models negotiate an optimal interpretation of the analyst interaction. This roughly follows the “Iterative, Alternating Algorithms” projection pipeline from the previous chapter in the interaction direction.

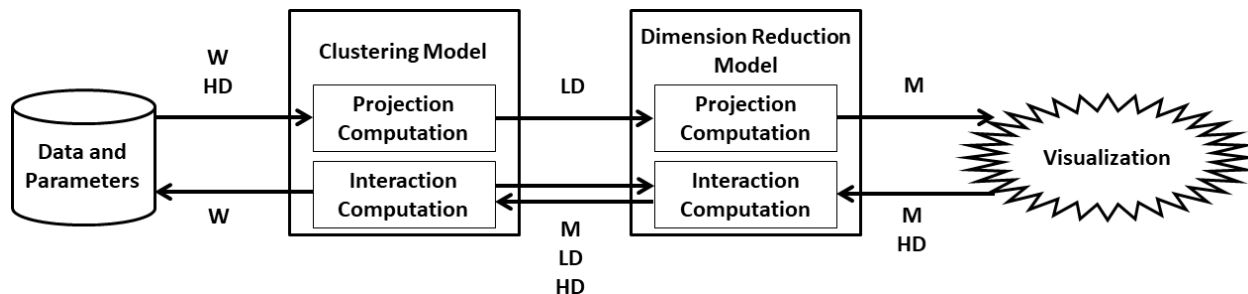


Figure 5.11: A representation of data flow in which the Interaction Computations of the Dimension Reduction and Clustering algorithms negotiate an optimal interpretation of the analyst interaction.

5.4 Interactions on Clusters

In this section, we consider the possible interpretations that result when an analyst repositions a cluster in the projection. We begin by describing cluster movement interactions that communicate a similarity relationship to other observations or clusters, and follow that discussion with interactions unique to the cluster-to-cluster relationship. We summarize the properties of these interactions in Table 5.5.

5.4.1 Repositioning Interactions

Much like repositioning an observation with respect to another component of the visualization, repositioning a cluster to a new position is most naturally handled by the Dimension Reduction Model. The right column of Table 5.2 summarizes some cluster intents and inter-

Table 5.5: A summary of the cluster interactions discussed in this paper, again evaluated by cardinality, the importance of the interaction source, target, or both, and whether an analyst is typically thinking high-dimensionally, low-dimensionally, or both.

	Cardinality	Source/Target	High-D/Low-D
Cluster Change in Membership			
Move cluster into cluster	1:1	T	HD
Move cluster out of cluster	1:1	S	HD
Move cluster between clusters	1: n	B	HD
Move cluster external to clusters	1: n	B	HD
Move cluster within a cluster	1:1	B	HD
Join/Split Clusters			
Join Clusters	n :1	T	HD
Split Cluster	1: n	T	HD
Create/Remove Clusters			
Create Cluster	1	T	HD
Remove Cluster	1	S	HD

actions with respect to both observations and clusters. In general, the Dimension Reduction Model is most suited to measuring and responding to interactions that cause distance changes between clusters and other visualization components.

However, a significant difference between observations and clusters is the space taken up by each in the projection – observations require a single point, while clusters require more space. As a result, a visualization designer should consider how to compute the location and value of a cluster in these interactions. Such computations could consider a simple centroid of the cluster, or potentially a weighted centroid based on the position of each observation within the cluster and some additional parameters. Further, distances between a cluster and another component of the visualization could be computed in a variety of ways, including single linkage, average linkage, and complete linkage [289]. Given that a cluster has a complex value that could be determined in a variety of ways, determining how to update weights based upon the result of an interaction is also a complex computation.

There is further ambiguity with respect to drag interactions on clusters, particularly in the case where cluster membership is encoded by boundaries. A drag interaction on such a cluster could be implemented so that all observations contained within the cluster are also repositioned with the cluster. This may be the most natural interpretation expected by an analyst. Such an interaction would be communicating a similarity relationship and handled by the Dimension Reduction Model, with the interaction computation of the Clustering Model skipped since no clustering reassignments have occurred. Such an example is provided in Figure 5.12. Alternatively, the analyst may merely be performing an exploratory interaction to learn relationships between the interacted-upon cluster and other components in the projection. In that case, no interaction computations are necessary.

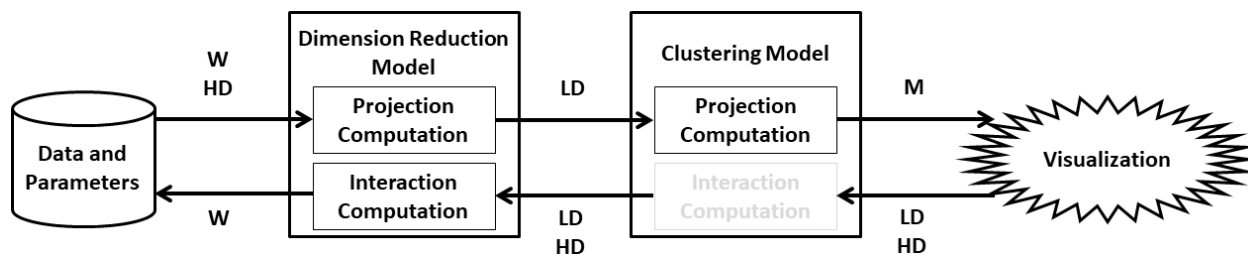


Figure 5.12: A representation of data flow for an interaction that repositions a cluster to another location in the projection, only requiring the interaction computation of the Dimension Reduction Model.

However, it is also possible to use a drag interaction to reposition the boundary of a cluster without relocating the observations that it encloses. Such an interaction could be used by an analyst to correct for misclassifications, encapsulating additional observations within the cluster by shifting the boundary. This interaction should be interpreted by the system with a very different meaning, as the analyst is again performing an expressive interaction to reclassify observations. The cluster membership reclassifications are then handled by the Clustering Model, as shown in Figure 5.13.

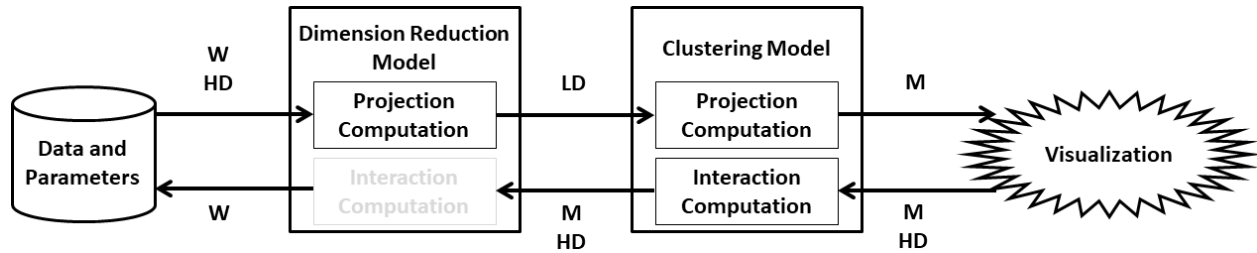


Figure 5.13: A representation of data flow for an interaction that repositions cluster boundaries to encapsulate new observations, only requiring the interaction computation of the Clustering Model.

5.4.2 Additional Cluster Interactions

In addition to the repositioning relationships that can be implemented for clusters, a further set of cluster-specific interactions are possible to implement through ambiguous operations upon a projection. For example, consider the interaction in which an analyst drags one cluster towards another until their boundaries overlap. One potential interpretation for this interaction is that the analyst is again providing a similarity relationship, indicating that these clusters are similar and again handled by the Dimension Reduction Model.

On the other hand, the analyst could be intending that the clusters be merged into a single, larger cluster. Alternatively, if the analyst drags one cluster fully into another, they may be demonstrating an intended hierarchical relationship between these two clusters. Both of these interactions are naturally handled by the Clustering Model, necessitating that the interaction computation of the Clustering Model determine which of these analyst intents is best matched by the interaction. Indeed, the interaction computations of the Dimension Reduction and Clustering Models require some internal communication and negotiation to jointly determine the intent of the analyst, again returning to the pipeline model of Figure 5.11.

Such ambiguity within a single interaction computation and across interaction computations is not limited to joining clusters. For example, consider a sequence of interactions in which

an analyst drags some nodes to the left side of a cluster and others to the right side. In one possible approach, such an interaction is interpreted as exploratory, and as such is not handled by either interaction computation. However, the analyst could also be indicating an intent to split this cluster into two (or more) smaller clusters. In this case, both the clustering and dimension reduction interaction computations are necessary: the clustering to create the new clusters and update assignments, and the dimension reduction to examine the dissimilarities between observations communicated the groups that the analyst formed to determine the appropriate cluster split and further update weights globally. Again, the pipeline model of Figure 5.11 with communication between the dimension reduction and clustering interaction computations is necessary.

In addition to these interactions, an analyst could also attempt to create a new cluster by repositioning a set of observations into a single region of the projection. Again, the interaction computations of the Dimension Reduction and Clustering Models would need to communicate, as the analyst is creating a new cluster while also communicating similarity relationships amongst the collection of points that they group together. Other interactions such as removing clusters, growing or shrinking the size of a cluster, and increasing or decreasing the importance of a cluster could be implemented through ambiguous interactions that must be interpreted to understand analyst intent.

Resolving the ambiguity in these interactions is more complex, as nearly all of the analyst intent is internal to the interaction computation of the Clustering Model. As such, providing additional visual feedback to communicate the system's interpreted intent can help to resolve issues that can result from these interactions. We discuss this further in the next section.

5.5 Discussion

In this section, we briefly summarize some additional considerations that a system designer should factor in when designing a system that incorporates an interactive projection with ambiguous interactions. We also discuss interactions that are useful for such applications but do not fall within the scope of the “with respect to what” problem, and present future research opportunities in this interaction design space.

5.5.1 Visualizing the Feedback

As an analyst performs these interactions in a projection, some of the ambiguity can be removed by providing the analyst with visual feedback demonstrating how the system will interpret the interaction. Such features are similar to those seen in Explainable AI systems [136] as they reveal details of the underlying model state. Tools with interactive projections have already implemented such feedback techniques. For example, StarSPIRE [38] includes an automatic text highlighting feature, more clearly displaying words judged to be important by the underlying models. This feature is used by analysts both to determine the overall importance of a document within the projections and to locate the important phrases and sections of a document [319]. In contrast, Andromeda [274] uses a dynamic-length slider to indicate the weights applied to dimensions. Related techniques seen in visual interfaces include changing the color of observations and drawing boundaries around clusters.

5.5.2 Shared or Separate Weight Vectors

Much of the discussion in the sections above has only briefly touched on the precise changes made to system parameters after handling an interaction. In single-model systems like

Andromeda, the weight update is straightforward as there is only one model learning weights. In multi-model systems such as StarSPIRE, the weight update becomes moderately more complex, as a relevance threshold needs to be learned in addition to set of term weights. Further increasing in complexity, SIRIUS [98] uses two separate weight vectors, one for the observation projection and one for the attribute projection, each of which are computed separately but with some dependency between them as interactions are performed.

Thinking more generally about a visualization system that incorporates both dimension reduction and clustering algorithms, a designer should consider whether each model should maintain its own weight vector or if a weight vector should be shared between models. The tasks supported and pipeline selected both play a large role in this decision. For example, the tool introduced in the next chapter follows the “Dimension Reduction Preprocessing for Clustering” projection pattern, with the cluster assignments naturally following from the low-dimension positions of the observations. In such a case, it is natural to support a shared weight vector between the models. In contrast, a system like iVisClustering [196] which computes dimension reduction and clustering separately on the high-dimensional data without interaction between the two (the “Independent Algorithms” pattern from the previous chapter) naturally supports separate weight vectors.

5.5.3 Parametric Interactions

In contrast to the interactions discussed in previous sections, Self et al. also defined a class of *parametric interactions* [273]. These parametric interactions provide explicit instructions to the system, bypassing the learning step necessary in the interaction computations and setting a precise value for a weight or other parameter. Though a different class of interactions entirely, these interactions are still quite useful in visualization systems. Self et al.

further identified a collection of low-level analytical tasks in Andromeda that are solved more efficiently by parametric interactions, particularly in interactions that focus on identifying values of a small number of dimensions [274]. In addition to the slider bars in Andromeda, a variety of techniques have been implemented to afford this functionality, including but not limited to Star Coordinates [174] and SpinBox widgets [10]. These parametric interaction techniques work equally well with projections of observations or attributes [98, 303].

5.5.4 Design Considerations

A visualization designer should consider the following dimensions when interpreting the intent of an interaction:

- **Interaction Target:** The interaction could be applied to the observations, the clusters, or both.
- **Cardinality:** The interaction could be applied to a variety of cardinalities: the nearest observation or n observations, all observations within a cluster, or all observations in the projection.
- **With Respect To What:** Is the relationship relative to the interaction at the source location, the destination location, or both?
- **Level of Thinking:** When performing the interaction, is the analyst thinking high- or low-dimensionally? In other words, is the analyst merely altering the projection, or are they considering all properties of a group of observations?
- **Visual Design:** Is the intent of the interaction influenced by the way that observations and clusters are encoded in the visualization? For example, using a boundary to delineate cluster membership may imply that dragging an observation across the boundary leads to a reclassification.

5.5.5 Towards Resolving Semantic Interaction Ambiguity

Semantic interaction aims to improve the quality of user interactions by enabling an analyst to directly manipulate a projection rather than attempt to finesse the parameters of the underlying mathematical model(s) [97, 110, 114]. However, this chapter demonstrates that the variety of possible meanings and intents of an analyst’s interactions can be difficult to capture in a single tool. In other words, interactions such as repositioning an observation are inherently ambiguous; this is the “with respect to what” usability challenge [273]. Introducing clusters can make some interactions easier by introducing a hard target, but also introduces added ambiguity (e.g., has the analyst moved an observation into a cluster, or was their goal to move the observation closer to some of the observations within the cluster?).

Resolving this ambiguity is critical to the future of semantic interaction. As such, a number of techniques have been introduced to provide feedback to the analyst regarding how the system will interpret their interaction [158]. For example, Figure 5.6 displays the selection interactions in Andromeda, including nearest neighbor selection, radius selection, and additional observation selection. The tools that will be introduced in the next three chapters also limit the interaction space to reclassifying observations and manipulating their position within clusters. However, limiting the interaction space can prevent analysts from learning more about their data from forbidden interactions.

To truly allow for free-form interactivity and data manipulation in systems, there is an inherent tradeoff between creating complex interactions that are precise but difficult for analysts to remember and perform and creating simple interactions that are ambiguous but easy for analysts. Precise interactions could include components such as a double-click to indicate the importance of the source of the target of the update, multitouch to denote the cardinality of the interaction, and presenting visual feedback to the analyst before the

interaction is handled by the system [61, 246, 325]. More ambiguous interactions could learn from a small training set and/or the interaction history to match the intent of a user to the interactions that they perform, such as found in ActiveInk [257]. Such a training set could be generated by an elicitation study, understanding precisely how analysts wish to perform these interactions.

5.6 Conclusion

This chapter models the complexity and ambiguity inherent in the interaction space of dimension reduction and clustering algorithms in interactive projections. We framed this discussion in the context of the “with respect to what” problem, an open research challenge in visual analytics identified by Self et al [273]. Through our discussion, we identified several factors necessary to consider for such interactions: thinking in high- or low-dimensional space, interaction with observations or clusters, interaction with source and destination, and cardinality of interaction. We presented a series of pipeline representations that incorporate both a projection direction to generate the visualization and an interaction direction to handle the interaction and interpret the intent of the analyst. Finally, we discussed additional considerations related to the implementation of such systems, as well as supplementary interactions and visual metaphors that further assist in communication and exploration.

Chapter 6

Castor: Dimension Reduction First

Semantic interaction is a direct manipulation technique that supports sensemaking by allowing users to actively manipulate the layout of data within a visualization. Through these interactions, an underlying model is updated, effectively letting the visualization system learn the intentions of the user by interpreting their actions as either *exploratory* or *expressive* [109]. Previous work has shown that coupling semantic interaction with data visualization is helpful for data exploration [157].

In this chapter, we propose, implement, and evaluate an interactive projection designed for quantitative data, which explicitly defines clusters within the layout of the data. In this model, the direct manipulation of nodes internal to these clusters represents the *exploratory* semantic interactions, while manipulations into and out of clusters represents *expressive* interactions. Through the expressive interactions, a feedback loop develops between the user and the layout algorithm. User manipulations with nodes and clusters suggest intentions regarding the importance of data attributes to the layout algorithm, which in turn updates the layout algorithm and hence the clusters with this information. Iterating through this sensemaking feedback loop provides insight to the user. In addition, explicitly defining the clusters in this model affords a partial solution to the “with respect to what” problem, as discussed in [273] and described in the previous chapter.

Our contribution in this chapter is the proposal, implementation, and evaluation of this dimension reduction-first model. Though previously-developed tools include explicit data

clusters and semantic interactions, while others include dimension reduction and clustering in the same projection, this is the first system to include all three: explicitly define clusters and cluster interactions for semantic interaction in a dimension-reduced projection. Our intention here is to explore one solution in the clustering/dimension reduction/semantic interaction design space.

6.1 Development and Design

This chapter is the first of three to introduce tools that implement ideas from the surveys in Chapters 4 and 5. In this section, we briefly discuss our development process in Section 6.1.1, after which we describe the similarities and differences between these three tools and briefly comment on alternatives in Sect 6.1.2. In summary, these tools each present both dimension reduction layout and cluster membership information within the same projection, while learning new dimension weights and hence new projections and clustering assignments based upon user interactions within the projections. The tools differ only in the algorithm interactions within the computational pipelines that generate the visualizations. We opted to keep the individual algorithms, visual style, interactions, datasets, and learning behavior identical across the collection of tools.

6.1.1 Development

We applied an iterative user interface design process [230] when creating these tools, though the primary focus of this process was during the development of the first tool (though we discuss them in a different order in the next section, the dimension reduction-first tool Castor was the first that we developed). The final design for Castor was modeled in the vein of

Vizster [148], which uses a force-directed layout algorithm to position observations within the projection and encases clusters within convex hulls in some community structure views.

During the development process of Castor, we received several rounds of feedback, including both individual feedback and group feedback from a research group environment. This feedback touched on both the visual design, the interactions, and the tasks to be supported by the system¹, and was delivered by both students and faculty from both computer science and statistics perspectives.

After the design for Castor was refined and published [317], we began work on both Pollux (Chapter 7) and Gemini (Chapter 8) concurrently. Because we developed Castor with the goal of extending the computations to other systems, we made use of standard code modularity practices and the rapid prototyping pipeline ideas of Dowling et al. [97] to reuse much of the Castor infrastructure in developing Pollux and Gemini.

6.1.2 Similarities and Differences Between the Tools

Same algorithms: Because we wish to explore the effects of changing the processing pipeline, we use the same dimension reduction and clustering algorithms in all tools. Dimension reduction is handled by the use of a force-directed graph. Clustering is performed using weighted k -means, learning the optimal number of clusters dynamically using the elbow method [296].

Both algorithms use approximately the same Euclidean distance function, the Pollux also introduces the idea of edge class weights. We compute a distance $\delta(n_i, n_j)$ between each pair of observations n_i and n_j by summing over all attributes $a \in attr$, with an associated

¹For example, early prototypes for Castor were focused on interactive cluster join and split operations, with clusters encoded using glyphs rather than convex hulls. We moved away from this direction because of the feedback that we received, which indicated that observation-centric tasks were more interesting to explore than cluster-centric tasks.

weight w_a applied to each attribute to denote the importance of that attribute to the current projection. When each system is initialized, each of the weights are set to 1, indicating that each weight has no larger or smaller effect on the resting length of each link than any other weight. Future weights are learned from user interactions with each system.

Both the force-directed layout and k -means clustering algorithms were selected for their computational efficiency relative to other algorithms in their families, to better enable real-time interactions for datasets up to several hundred observations and dimensions in size. There is a wide breadth of other algorithms that we could have selected, not just limited to those summarized in Chapters 2 and 3. Many recent tools use either PCA or MDS for dimension reduction, though k -means remains the default choice for clustering. Indeed, a developer could substitute these algorithms into the tool pipelines described in Section 4.2.1 without changing the overall data flow, though the precise clustering and projection results yielded will certainly change.

Same visual style: We also maintain the same visual style for each of the tools. Individual observations are represented by circular nodes, changing colors based on whether or not they are fixed in place. Cluster centroids are represented by red squares. Both observations and clusters have an accompanying label positioned either above or below. Clusters are represented by a shaded convex hull that surrounds all observations in that cluster. A list of all dimensions and their accompanying weights are listed on the right side of the interface.

There are a variety of alternative methods for visualizing cluster membership in addition to the convex hulls that we selected. Color encodings to indicate cluster membership appear often in the literature [9, 74, 129, 196], as does position [65, 221]. Each of these three techniques makes use of preattentive processes for quick recognition of clusters [147]. Dual-encoding of cluster information is also often seen, including recent tools such as TopicLens [181] and UTOPIAN [63]. Observations have fewer options, but some systems have been developed

to expand nodes into documents that can be highlighted [38], while others use images to display descriptive information about a node [56].

Same interactions: Similarly, we chose to maintain the same set of possible interactions in each tool. Hovering over a node will display details-on-demand information, listing the high-dimensional data of that observation. Clicking on a node fixes it in place within the projection, changing its color from black to bright green to signify its new control point status. Dragging a node will reposition it in the projection, automatically fixing it in place at the end of the drag, and will influence the position of other nodes. Dragging a node across a cluster boundary will trigger the learning routine. These final two interactions are most important to interactions with the computational pipelines: dragging a node across a cluster boundary is treated as a reclassification interaction that executes the Clustering Model, while a drag action that does not cross a cluster boundary is treated as an intended change in similarity that executes the Dimension Reduction Model. We note that these two interactions are treated in our tools as mutually exclusive blocks, though this does not need to be the case. We speak more about the need for a more complex interaction space in Sect. 8.4.

While the introduction of clusters into a projection also allows for a variety of new cluster-centric interactions (including cluster merging, splitting, and creation [63], cluster annotation [175], and cluster hierarchies [228]), our development process feedback indicated more interest towards observation-centric interactions.

Same learning behavior: Much as with our choice of force-directed layout and k -means clustering for computational efficiency, we use a heuristic learning approach to efficiently approximate an inversion of the distance function. For example, assume that a node is dragged into a cluster. The inference made by each system is that the current exploration strategy of the analyst has included this dragged observation with the others in the cluster

because of some shared properties of interest to the user. By measuring the difference between the value of the cluster centroid c and the dragged node n for each attribute a , we create a sorted list of which dimensions are similar and which are dissimilar.

Note that here we use L_1 or Manhattan distance rather than Euclidean distance because we consider each attribute independently with the goal of sorting them rather than considering the attributes collectively to calculate an overall distance. After sorting the dimensions, the similar dimensions should then influence the next iteration of the projection more strongly, while the dissimilar dimensions should have less influence. Therefore, the weights of the similar dimensions are reduced (to draw nodes closer together in the projection), while the weights of the dissimilar dimensions are increased. This set of dimension weights is shared by both algorithms in each tool.

A variety of methods could be used to replace our heuristic learning method, including those that are mathematically rigorous [274], probabilistic [157], or even an identity function in some cases [97]. With the exception of the identity function, these solutions are more computationally complex than our heuristic, which can make the display noticeably slower in a system with real-time animation updates [38].

6.2 Model and Implementation

Our model of this cluster-based semantic interaction framework is shown in Figure 6.1. This bidirectional pipeline is divided into forward and backward directions, where the forward direction handles the data projection and the backward direction responds to user input. Moving forward (right) along the pipeline, data is represented in a node-link diagram where link lengths encode similarity. A force-directed algorithm places the nodes to minimize force exerted by the links as they converge to their resting weights, while a modified k -means

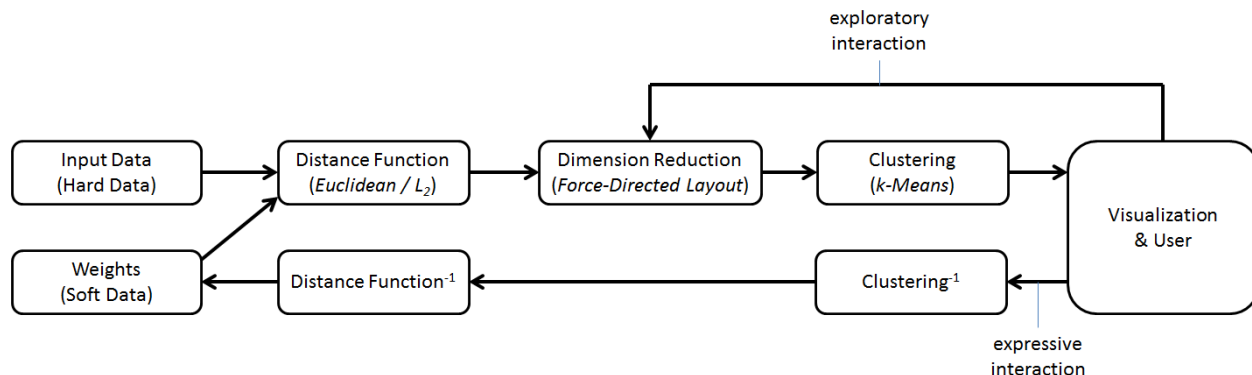


Figure 6.1: The framework and implementation of our cluster-based semantic interaction model.

algorithm determines clusters in the data. Users then interact with the visualization, either through *exploratory* interactions which solely adjust the visualization (the force directed layout and forward) or through *expressive* interactions which follow the pipeline backward. In this case, the system determines the clusters affected by the interaction, computes similarity measures to learn which weights to update, and changes the weights associated with attributes in the layout, leading to the next forward flow along the pipeline. The following implementation makes use of the Euclidean distance function, a force-directed layout for dimension reduction, and k -means clustering; however, the model generalizes to any distance function, dimension reduction technique, and clustering algorithm. The following subsections detail the computational processes captured by each node in Figure 6.1.

6.2.1 Projection Direction

Distance Function: The data projection begins with computing a distance $\delta(n_i, n_j)$ between each pair of data items n_i and n_j . This distance, described in Equation 6.1, is the Euclidean or L_2 distance between the normalized attributes of n_i and n_j , including a weight w_a applied to each attribute a to denote the importance of that attribute to the current

projection. At system initialization, each of the weights associated with the dimensions in the dataset are set to 1, indicating that each weight has no larger or smaller effect on the resting length of each link than any other weight. These weights are updated in response to user interaction in the interaction direction to create new projections, detailed in the next subsection.

$$\delta(n_i, n_j) = \sqrt{\sum_{a \in attr} w_a * (n_{i,a} - n_{j,a})^2} \quad (6.1)$$

Force-Directed Layout: Once a distance is computed for every data pair, we load the data into a force-directed node-link visualization such that every data item n_i is encoded as a node and every distance $d(n_i, n_j)$ between pairs of data items is encoded as a link with the distance value mapped to the resting length of the link. To address the nondeterministic layout challenge inherent to force-directed placement, node positions are initially set to the same location every time the system is initialized. The nodes begin at locations uniformly and radially spaced about the center of the display. The force-directed layout algorithm is run repeatedly until it converges to a relatively stable layout, at which time we begin to visualize the layout.

Clustering: As the force-directed layout converges to a solution, we begin to draw clusters surrounding groups of nodes. These clusters are computed using a modified k -means algorithm, which has been altered to include a maximum cluster radius that allows some nodes to exist external to all clusters. Similarly to the initial node placements, cluster centroid positions are initialized uniformly and radially spaced about the center of the display, and converge towards a final cluster layout as the force-directed algorithm continues to update the node positions. The k -means clusters are kept stable by seeding the new clusters with the previous centroid positions each time the display refreshes. We selected k -means as an efficient and simple clustering algorithm. The user is afforded control of both k and the size

of each cluster in pixels from the centroid (the system defaults are $k = 5$ clusters and a 200 pixel radius for each cluster). Clusters are drawn using the Graham scan algorithm for convex hulls [135].

Visualization & User: The user does not need to wait until a final layout of nodes and clusters is reached before beginning to interact with the system. They can begin performing interactions from the moment that the layout begins to render on the display, which triggers the backward direction of the pipeline as described in the next subsection.

6.2.2 Interaction Direction

The user interacts with the nodes via direct manipulation, using click-and-drag actions to move nodes around the screen and mouse over interactions to see the details of a node. Keyboard inputs allow the user to change the number and size of clusters. As noted in the introduction, users can perform both *exploratory* and *expressive* interactions in semantic interaction systems. These interactions take two different backward paths through the pipeline.

Exploratory interactions are defined as interactions that alter the layout of the projection but do not affect the model. In contrast, expressive interactions will alter the model, in turn also affecting the layout of the projection. To address the “with respect to what” interaction challenge, we separate exploratory and expressive interactions based upon their effects on the clusters.

Exploratory interactions are interactions with nodes in which the user does not move a node into or out of a cluster; in other words, moving a node internal to a cluster to a new location in the same cluster, or moving a node external to all clusters to a new location external to all clusters. We note that it is possible that an exploratory interaction will cause nodes

to shift membership of nodes between clusters in response to this interaction. However, these membership changes are due to the force-directed layout rather than any change in underlying model parameters. Additionally, moving nodes will pin their location on the display, which forces a further alteration to the force-directed layout so that links support the new position of the node. Again, this does not change the distance model parameters.

Clustering⁻¹ (handling cluster interactions): We define three expressive interactions that update weights in the graph: adding a node from the external region to a cluster, removing a node from a cluster to the external region, and transferring a node from one cluster into another. The third interaction, transferring a node between clusters, is interpreted as a sequence of the first two actions, removing the node from the first cluster and then adding it to the second cluster. Thus, we will discuss only the insertion and removal processes in detail. The goal of each of these interactions is to learn the projection and clustering parameters based on the intent of the cluster assignments that the user performs, thus creating a new projection and set of clusters that reflect the user action.

After an observation is reclassified and the weight vector is recomputed, the new weight vector is applied to all observations. The updated positions of the nodes can thus result in cluster membership updates for some observations. In the case shown in Figure 6.2, reclassifying the Grizzly Bear into the “Predators” cluster has updated the cluster membership assignments of four other observations: The Rat, Raccoon, Weasel, and Polar Bear. The analyst can identify these observations from the blue colored nodes, with lines connecting them to the centroid of the cluster from which they departed.

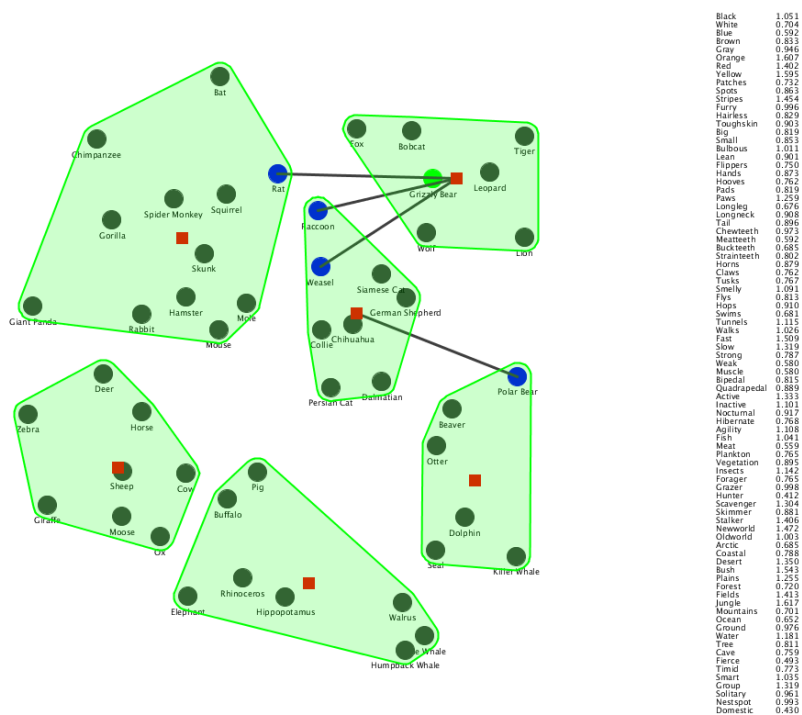


Figure 6.2: After reclassifying the Grizzly Bear into the upper-right “Predators” cluster, the Rat, Raccoon, and Weasel all were reassigned out of the “Predators” cluster, and the Polar Bear was removed from the “Pets” cluster.

Adding Nodes to Clusters

Distance Function⁻¹ (handling projection interactions): If a node is moved from the external region into a cluster, we aim to understand why the user decided that the node belongs to this cluster by analyzing what attributes in this node are similar to those attributes present in the cluster, and also what attributes in this node conflicts with in the cluster. To make this judgment, we use a heuristic approach to efficiently approximate an inversion of the distance function. We compare each attribute a of the cluster centroid c with the corresponding attribute of the newly added node n . This comparison is a calculation similar to that of our initial distance computation, normalizing the difference in value for each attribute between node and the cluster centroid, and is shown is Equation 6.2. Note that here we use L_1 or Manhattan distance rather than Euclidean distance, because we consider each

attribute independently with the goal of sorting them rather than considering the attributes collectively to calculate an overall distance. The motivation behind this computation comes from the user interaction – the user has decided that node n belongs to cluster c , and so the attributes that are most similar between the node and the cluster are important to the user. Therefore, the model should reflect the importance of these attributes in the visualization. Similarly, the attributes that are most different between node and cluster are irrelevant to the user and hence less important to the model.

$$\forall a \in attr, \delta(c_a, n_a) = |c_a - n_a| \quad (6.2)$$

After this similarity analysis is complete for each attribute, we sort the attribute collection based on the strength of similarity score computed, with the sorted positions of attributes with tied similarity scores placed arbitrarily. A function is then applied to each of these sorted attributes to update the weights of each of the attributes, so that attributes that show the greatest similarity pull nodes closer together while attributes that do not push nodes further apart. The attributes that display strong similarities between the node and the cluster have their weights reduced, so that when the link lengths are recalculated, nodes that have similar values for this attribute are pulled closer together. Similarly, attributes displaying difference between the node and the cluster have their weights increased to expand link lengths after the recalculation phase. Attributes near the middle of the pack have little change, with weight updates close to 1.

Following the weight updates, the system proceeds through the forward direction of the pipeline again, recomputing distances and resting lengths and updating the layout and clusters accordingly. As the layout is stabilizing, new insights can be drawn about the properties and layout of the nodes, and the system is ready for new interactions.

We note that it is possible for a node already positioned inside of a cluster to be forced out of the cluster because of a subsequent interaction. This is both because the system learns gradually rather than immediately and because we do not enforce must-belong and must-not-belong cluster membership constraints unless the nodes are pinned in place. A number of user-driven cluster assignments are required for the system to converge to a user-intended ideal projection. We chose to allow the system to update after every user interaction (rather than to allow a number of interactions followed by an “update layout” trigger) to allow users to immediately begin to learn and draw insights from individual interactions.

Removing Nodes from Clusters

Distance Function⁻¹ (handling projection interactions): If a node is dragged from a cluster into the external region between clusters, we aim to understand why the user decided that this node does not belong to its assigned cluster through a similar heuristic to the Add Node procedure. Again, we compare the attributes of the centroid of the cluster with the attributes of the removed node and sort the scaled outcomes.

Following this sorting, we again update the weights on the attribute collection based on the order of similarity scores. However, attributes that display strong similarities between the node and the cluster now have their weights increased to expand the length of links (interpreting formerly strong similarities as attributes that the user is not interested in clustering), while attributes that display differences between the node and the cluster have their weights reduced to shrink links (noting that this attribute is more important to the user than it was in the previous layout). Again, after all weights have been updated, new resting lengths are computed for each link, and the force-directed algorithm shuffles nodes into a new layout, with cluster membership updating as necessary.

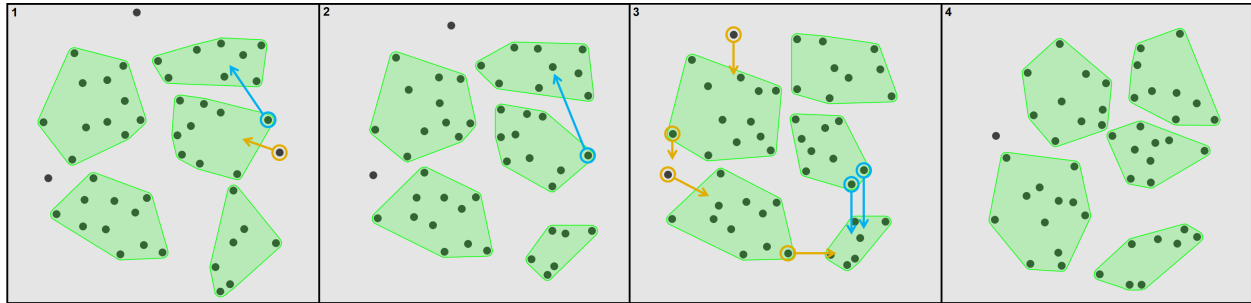


Figure 6.3: Four steps through the interaction discussed in the Usage Scenario section. Interactions initiated by the user are shown with blue arrows, while the cluster reassignments initiated by the system are shown with orange arrows.

6.3 Usage Scenario

Here, we show one example of exploring a dataset via cluster-based semantic interaction. The dataset used in this usage scenario is a collection of 49 animals, each defined by 85 numeric attributes that describe the animal’s color, physical characteristics, habitat, and behavior [195]. The data is loaded into the system using the default settings, creating five clusters with cluster membership size set at 200 pixels from the centroid. These clusters, along with the user actions described below, are shown in Figure 6.3.

Before any interactions take place, the similarity relationships already in the data have clustered 46 of the animals into groups with similar characteristics. Clockwise from the top, these groups can be summarized as Predators (including animals such as the Lion and Tiger), Pets (Chihuahua and Persian Cat), Aquatic animals (Dolphin and Seal), Large Grazers (Buffalo and Sheep), and Small Foragers (Skunk and Rabbit). Three animals (Bat, Polar Bear, Zebra) were not assigned to any cluster. There are also several animals that are misclassified according to this clustering interpretation, including the Grizzly Bear in the Pets cluster and the Giant Panda in the small foragers cluster. For the purposes of clarity and mental map stability in this usage scenario, we perform the exploratory action of fixing one node from each cluster in place, an action that has no effect on the underlying weights.

In exploring the data, one might begin by resolving the pet Grizzly Bear issue by removing the grizzly bear node from the pet cluster and placing it in the predator cluster (Frame 1 interaction). This combination of two cluster transfers (out of pet, into predator) is an expressive action that results in a weight update. Following this action, the Fierce weight has dropped from 1.000 to 0.437, indicating importance to that attribute in the current exploration and pulling nodes closer together through its impact on the resting length of all edges. In contrast, the Orange color weight has increased from 1.000 to 1.651, indicating a lack of importance to that attribute and increasing the contribution of that attribute to the rest lengths of all links.

This action has absorbed the Polar Bear node into the pet class incorrectly, so the user can also move the Polar Bear node into the Predator cluster (Frame 2 interaction), an expressive action that further reduces the Fierce weight to 0.224. Because of these actions, the Otter has now moved into the Pet cluster and the Beaver was already there, so the user may perform expressive actions to drag those into the Aquatic animals cluster (Frame 3 interactions). After those interactions have completed (Frame 4), the Water weight has decreased to 0.263, and the Fierce weight has decreased further to 0.116. Other attributes that the model found important include 0.103 for meat-based diets and 0.531 for having hooves, while attributes such as nocturnal (2.259) and hairless (2.327) were not judged as important to the current exploration of the user.

Most importantly, without user interaction on these nodes, these actions had the effects of putting the Zebra into the Large Grazers cluster, removing the Giant Panda from the Small Forager cluster and moving towards Large Grazers, added the bat into the Small Grazers, and moved the Walrus from Large Grazers to Aquatic animals. The final state is shown in the right-most panel of Figure 6.3.

6.4 Discussion

A number of the weight updates and animal shifts through the interactions in the usage scenario are worth noting. For example, the Zebra transitioned into the Large Grazers cluster and the Giant Panda left the Small Foragers cluster despite no user interaction with these nodes. The weight that the system judged to be most important was neither Water nor Fierce but Domestic, with a 0.050 weight. Still, similar weights to Fierce such as Hunter were also reduced to small values (0.184). The weights judged least important to the user interactions are Slow (2.435), Fast (2.318), and Active (2.326), demonstrating that the system learned that the user was interested in the diet and habitat of these animals, not their speed.

It is also interesting to think about the relationship between clustering algorithms and dimension reduction algorithms. In a way, making cluster assignments is equivalent to 1D dimension reduction, noting that the cluster assignment is the primary dimension of organization. Our system goes a bit beyond that, still respecting the 2D position of the force-directed layout while also encoding an extra dimension of cluster membership.

6.5 Limitations

A limitation to the implementation of this framework is the solution we implemented to address the nondeterministic layout challenge: we always initialize nodes and centroids in the same locations. Because of this, the same clusters will develop every time, potentially biasing the user towards these clusters and hindering new directions of exploration. Altering this initial placement to be more random is a trivial solution to this source of bias.

6.6 Conclusion

In this chapter, we have proposed a cluster-based framework for semantic interactions. Our implementation of this framework includes a feedback loop between the user and the back-end model, in which the layout of the nodes informs the clusters, while the user interactions between the nodes and clusters update the weights that compute the layout. Though our specific implementation makes use of the Euclidean distance function, a force-directed layout, and k -means clustering, the overall model generalizes to any distance function, dimension reduction technique, and clustering algorithm. We show in a usage scenario an example of sensemaking using a dataset of animal characteristics, demonstrating that the model can learn from user interactions and affect the layout of nodes and clusters that are not interacted with directly by the user.

Chapter 7

Pollux: Clustering First

This chapter introduces Pollux. Pollux is similar to Castor, but the algorithm order is reversed: the data is first clustered in the high-dimensional space, and is then projected into a two-dimensional visualization. Such a process has previously been included in analytical tools [90], but these projects did not include semantic interaction learning as seen in Castor. Introducing similar online learning into Pollux permits analysts to maintain a data exploration focus, with no need for mental context switching to ponder model parameters. Determining how to present the outcome of this computational flow leads to a number of design options that can be considered, reflecting a balance between an accurate projection of the data and faster rendering.

In particular, we note the following contributions:

1. The design and implementation of Pollux, an interactive cluster-first system that learns observation classifications via analyst feedback and displays using a unified layout model based on edge classes.
2. A discussion of the benefits of the cluster-first model, as well as of methods that can extend the cluster-first design space beyond that which has been implemented in this work.

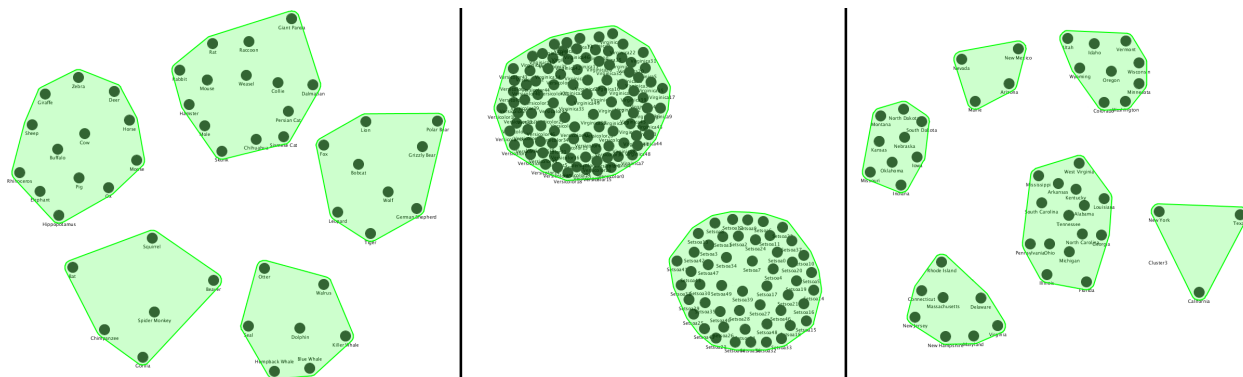


Figure 7.1: Clustered projections of three datasets generated by Pollux. From left to right, an Animals dataset [195], the Fisher Iris dataset [102], and a U.S. Census States dataset [305].

7.1 Pollux

The goal of the Pollux system is to continue to explore the interaction space between dimension reduction and clustering algorithms [320] by introducing a cluster-first system. Pollux differs from many other interactive clustering systems because of the inclusion of projections via dimension reduction, displaying learned similarities at both the cluster and observation level through user-driven reclassification (several examples are provided in Figure 7.1). An analyst using Pollux is afforded the ability to update the system-learned categorization of observation, training the underlying clustering and dimension reduction models to better express their current exploration interests. Further, an analyst should be able to receive feedback from these algorithms concurrent with the incremental learning process, permitting the analyst to update or alter their exploration based on the current results displayed.

Our model of this cluster-first framework is shown in Figure 7.2. This bidirectional pipeline is divided into projection and interaction directions, where the projection direction converts input data into an interactive visualization, and the interaction direction responds to analyst input. These projections and interactions are supported by Dimension Reduction and Clustering Models, which work cooperatively to generate an interactive visualization from

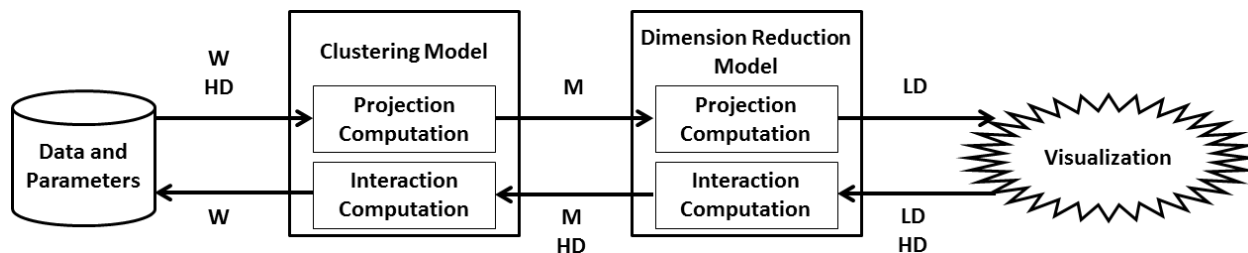


Figure 7.2: The computational pipeline for Pollux. The projection computations convert data into a visualization, while the interaction computations interpret and respond to analyst interactions.

the provided high-dimensional dataset and a learned weight vector. The implementation described in this section makes use of the Euclidean distance function, a force-directed layout for dimension reduction, and k -means clustering; however, the model generalizes to any distance function, dimension reduction technique, and clustering algorithm.

7.1.1 Projection Direction

At a high level, the projection direction computes clustering assignments for the high-dimensional observations, and then structures a visual representation of those clusters into a two-dimensional space. There are many methods to visually convey cluster membership, and again, we elected to follow the visual style of Castor for this application. A broader discussion of visualization structures and alternatives follows in Sect. 7.2.

Weighted Cluster Assignments In the projection direction of the Pollux pipeline, the Clustering Model is the first to execute. This cluster model has two primary goals: to determine a quality clustering assignment for the observations given the data at hand, and to communicate those membership assignments to the Dimension Reduction Model for layout.

To accomplish the first goal, a weighted k -means algorithm is executed on the dataset. In the default implementation of the system, we execute 500 versions of k -means for each value

of k ranging from 2 to 15. Each of the best k -clusterings (as determined by summed intra-cluster distance) is stored, and an optimal k value is determined from these best clusterings using the elbow method [296]. The analyst is afforded control of k , so that they can refine the number of clusters generated by the system if the initially-selected version does not suit their goals. After each of the clusters has been determined, an additional node is created to specifically represent the centroid of the cluster.

Projecting Clusters After cluster memberships have been determined, the Dimension Reduction Model is tasked with projecting these clusters into the visualization. The precise layout of this visualization is dependent upon the importance of each class of edges that is included in the layout. There are five of these classes, shown in Figure 7.3:

- *Centroid-Centroid Edges* (CC): Distances between the clusters themselves, displaying the similarity between pairs of clusters.
- *Centroid-Node Internal Edges* (CN_I): Distances between each cluster member and its centroid, demonstrating the centrality of a node in the cluster. These edges act to pull associated nodes towards their cluster centroid.
- *Node-Node Internal Edges* (NN_I): Distances internally between cluster members. These edges display the similarity of nodes within a single cluster, providing an overall organizational structure to the members of a cluster.
- *Centroid-Node External Edges* (CN_E): Distances between nodes and the centroids of other clusters. These edges pull nodes within a cluster towards the direction of alternative cluster memberships.
- *Node-Node External Edges* (NN_E): Distances globally between observations, with no regard for cluster boundaries (but not including edges internal to a cluster). These edges pull nodes directly towards similar observations in other clusters, showing pairwise relationships between observations.

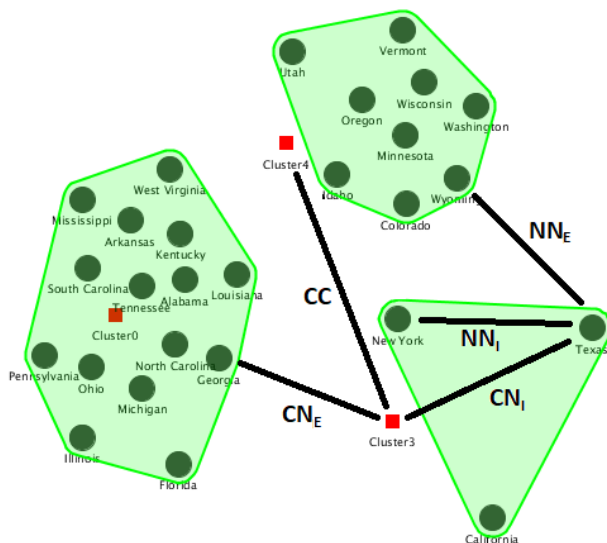


Figure 7.3: Five different classes of edges that could be included in the layout: Centroid-Centroid Edges (CC), Centroid-Node Internal Edges (CN_I), Node-Node Internal Edges (NN_I), Centroid-Node External Edges (CN_E), and Node-Node External Edges (NN_E).

The classes of edges that are included in the visualization impact both the accuracy and rendering speed of the visualization. A longer discussion of this tradeoff is included in Section 7.2.

After edges are constructed in the graph, a distance $\delta(n_i, n_j)$ is computed for every pair of nodes and centroids with a connecting edge. This distance, described in Equation 7.1, is the L_2 or Euclidean distance between the normalized attributes of endpoints n_i and n_j , including an attribute weight w_a applied to each attribute a to denote the importance of the associated dimension to the current projection. At system initialization, each of these weights are set to 1, indicating that each weight has no larger or smaller effect on the resting length of each link than any other weight. These attribute weights are updated in response to analyst interactions in the interaction direction, detailed in the next subsection. A further edge class weight, w_e , is applied to each of the edges. This edge class weight allows for different styles of visualization to be created (e.g., compact clusters, tightly grouped clusters, broad clusters). Tradeoffs in this design space are also discussed in more detail in Section 7.2.3.

These computed edge lengths are then treated as the optimal resting lengths within a force-directed simulation, with nodes beginning at locations uniformly and radially spaced about the center of the display and updating their positions until the layout converges to a relatively stable layout. Clusters are drawn using the Graham scan algorithm for convex hulls [135].

$$\delta(n_i, n_j) = \sqrt{\sum_{a \in attr} w_e * w_a * (n_{i,a} - n_{j,a})^2} \quad (7.1)$$

7.1.2 Interaction Direction

The goal of the interaction direction is to respond to analyst interactions, incrementally training the underlying models and learning the intent of the analyst when they perform reclassification interactions. The analyst interacts with the nodes via direct manipulation, using click-and-drag actions to move nodes between clusters. Mouseover interactions afford a details-on-demand view of the raw data for each observation (Figure 7.4). Analysts have the ability to perform two types of interactions, each of which are addressed by a different model in Pollux.

Layout Interactions These interactions are addressed by the Dimension Reduction Model, as no clustering updates need to be performed. Such interactions can be used to probe relationships internal to a cluster, perhaps dragging a node from one side of the cluster to the other and watching the updates to the rest of the layout. These interactions can also assist the force-directed optimization in Pollux to shift between various local minima in the layout of observations, and could be extended to navigating the rotation and scale invariance properties of other dimension reduction algorithms such as MDS. Performing such interactions will not trigger the learning of new attribute weights; these are only learned via expressive interactions.

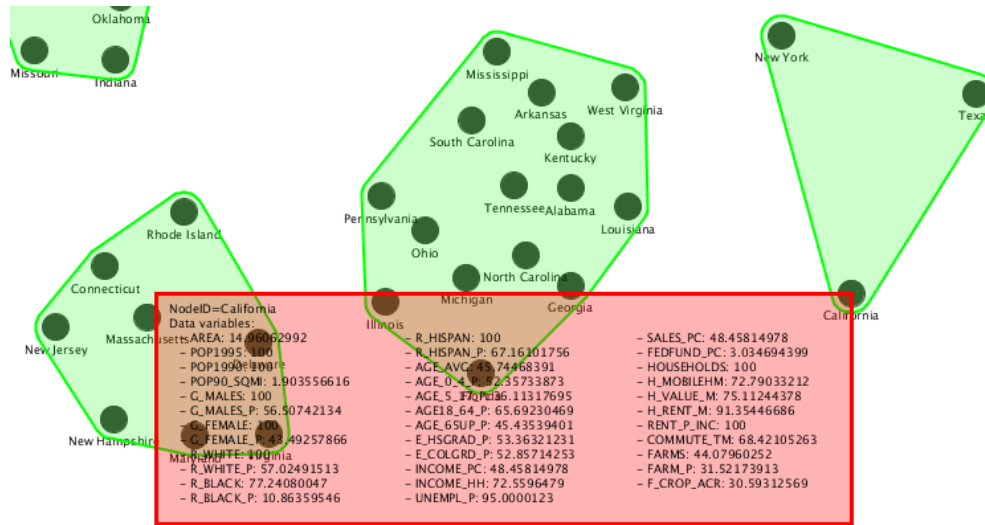


Figure 7.4: Mouseover interactions afford a details-on-demand view of the raw data for each observation.

Cluster Interactions These interactions are performed when an analyst reclassifies an observation, dragging it from one cluster into another. As this interaction demonstrates an analyst’s dissatisfaction with the automated membership assignment, the system begins to learn a distance metric that matches the exploration of the analyst, inferring the semantic reasoning behind this reclassification by examining the attributes of the dragged node, the source cluster, and the destination cluster. These interactions train the Clustering Model via incremental feedback, using a metric learning approach to efficiently compute an inversion of the distance function. We compare each attribute a of the source cluster centroid cs and the destination cluster centroid cd with the corresponding attribute of the dragged node n . As shown in Equations 7.2 and 7.3, this comparison is a calculation similar to that of our initial distance computation, normalizing the difference in value for each attribute between the node and cluster centroids. One important difference is that here we use L_1 distance rather than Euclidean distance, as we consider each attribute independently in order to sort them rather than considering the attributes collectively to calculate an overall distance.

$$\forall a \in attr, \delta(cs_a, n_a) = |cs_a - n_a| \quad (7.2)$$

$$\forall a \in attr, \delta(cd_a, n_a) = |cd_a - n_a| \quad (7.3)$$

After computing this similarity distance for each attribute, we sort the attribute collections based on the strength of similarity score computed, with the sorted positions of attributes with tied similarity scores placed arbitrarily. A linear function is then applied to each of these sorted attributes to update the weight of each attribute. Attributes that show the greatest similarity between cluster and node should pull the pair closer together and so the weight is reduced, while attributes that show the least similarity should push the pair apart and so the weight is increased. Attributes near the middle of the list have little weight, with weight updates only a fraction from 1. With these weight scaling factors set, we first apply the factors between source cluster and node, followed by those of the destination cluster and node.

After the attribute weights have been updated, the system must update the visualization through the projection direction of the pipeline again. First, cluster assignments for each node are recomputed with the new weight information. Any node that receives a new cluster assignment will have its adjacent edges updated as needed, which could include the removal of unneeded edges, the introduction of new edges, or the weighting of an edge that has transitioned from internal to external or vice versa. The force-directed layout then executes, with nodes that switch clusters smoothly animating from source to destination cluster.

Table 7.1: A summary of the three datasets visualized with Pollux, enumerating each edge type.

Dataset	Animals [195]	Fisher’s Iris [305]	Census [102]
Nodes	49	48	150
Dimensions	85	35	4
Clusters	5	6	2
Fully-Connected Node Graph	1176	1128	11175
Centroid-Centroid Edges (CC)	10	15	1
Centroid-Node Internal Edges (CN _I)	49	48	150
Node-Node Internal Edges (NN _I)	241	214	6175
Centroid-Node External Edges (CN _E)	196	240	150
Node-Node External Edges (NN _E)	935	914	5000

7.2 Extended Design Space

As noted at the beginning of Section 7.1, the Pollux model can be generalized to any distance function, dimension reduction technique, and clustering algorithm. Castor [317] held the same property. However, there are some additional properties of the Pollux technique that enable additional variants to be created from this cluster-first approach. In particular, we discuss in this section the roles of edge type selection and edge class weights on the visualization that is created.

7.2.1 Edge Class Selection

As noted in Section 7.1.1, there are five different classes of edges that exist within a Pollux projection. The role of these edges in the force-directed layout is the same as the role of distances in many dimension-reduction projections: to communicate a measure of similarity between two objects in the visualization. With the added introduction of clustering into Pollux projections, layout time can be improved and visualizations can therefore be generated more rapidly than in pure projection applications.

Table 7.1 provides a summary of nodes, dimensions, clusters (as learned by Pollux), and edge counts for three different datasets: a dataset of animals and a collection of appearance, habitat, diet, and behavioral attributes [195]; the traditional Fisher’s Iris dataset [102]; and a dataset of demographic, employment, and housing data for the 48 continental U.S. States [305]. In this table, the “Fully-Connected Node Graph” row provides the number of edges or distances required to lay out a fully-connected graph of only observations (this is merely the sum of the NN_I and NN_E categories). Each of these classes of edges communicate additional information to the analyst, as listed in Section 7.1.1.

7.2.2 The Effect of Edge Class Selection on Performance

By selecting only the CC , CN_I , and NN_I edge classes, the number of edges that need to be computed for a projection is reduced by approximately 25-50% for these datasets, while still displaying the relative similarity of both clusters and observations internal to those clusters. Though force-directed simulations can run as quickly as the $O(n \log n)$ of the Barnes-Hut simulation approximation [23], many force-directed simulation implementations are $O(n^3)$, yielding a significant performance boost with a reduction to half of the original number of edges.

Figure 7.5 shows Pollux layouts with default edge class weights for the Census dataset, displaying the visualizations generated from four different edge class selections. In Figure 7.5A, only the CC and CN_I edges have been selected. As a result, both pairwise cluster similarities (CC) and cluster memberships (CN_I) are displayed. In Figure 7.5B, the NN_I have been added. The added attractive forces between nodes internal to each cluster act to compact the clusters in most cases while also incorporating pairwise node similarities internal to each cluster. Figure 7.5C adds CN_E edges, which act to pull nodes within clusters towards the

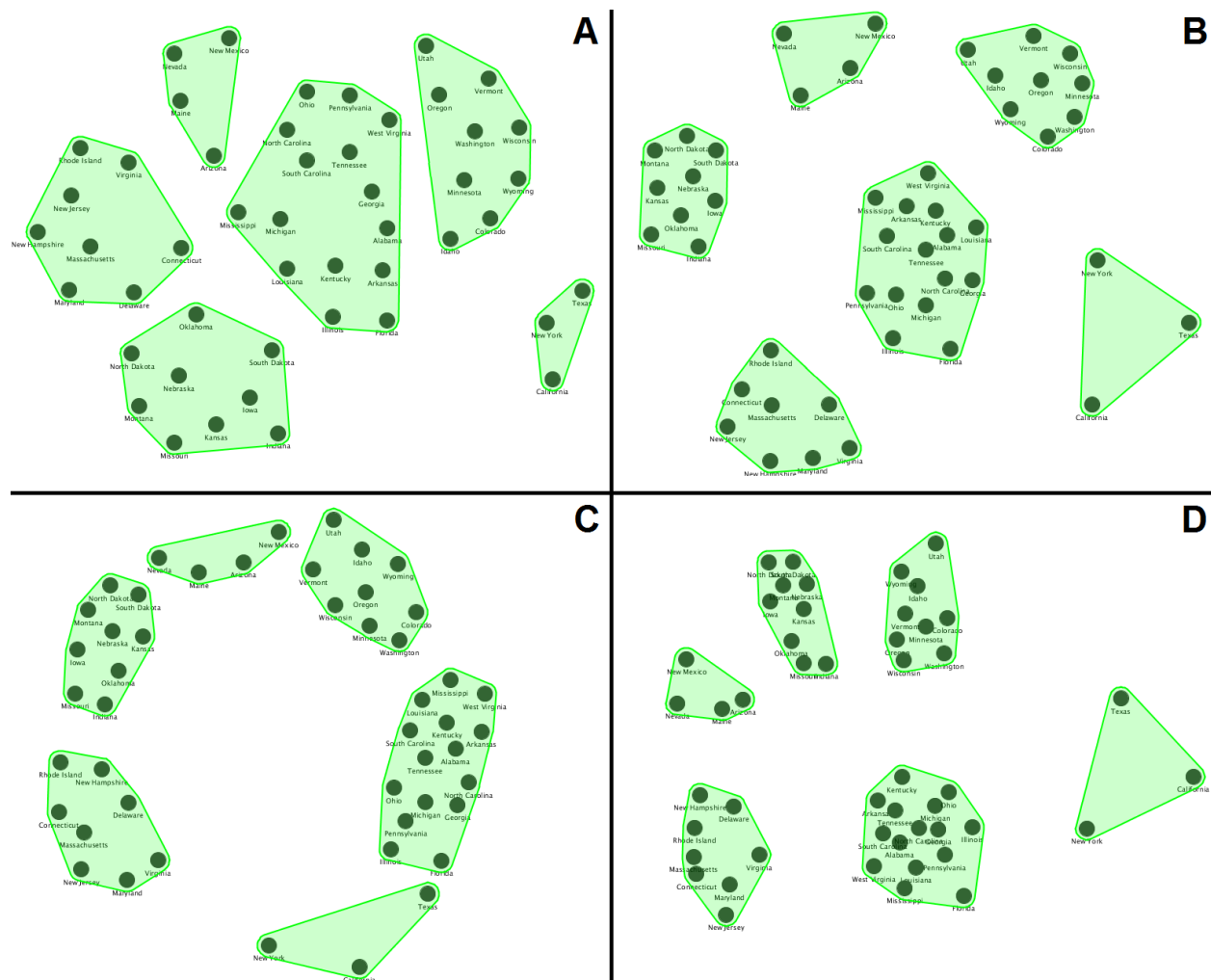


Figure 7.5: Four views of the Census dataset with a variety of edge class selections. From top to bottom, (A) only CC and CN_I edges, (B) same as above, plus NN_I , (C) same as above, plus CN_E , (D) all edge types.

centroid of other clusters, thereby causing a more radial layout with the nodes on the inner ring boundary most attracted to other clusters and those on the outer ring boundary least attracted. Finally, Figure 7.5D adds NN_E edges, which again cause the clusters to compact as many additional edges are added to the graph.

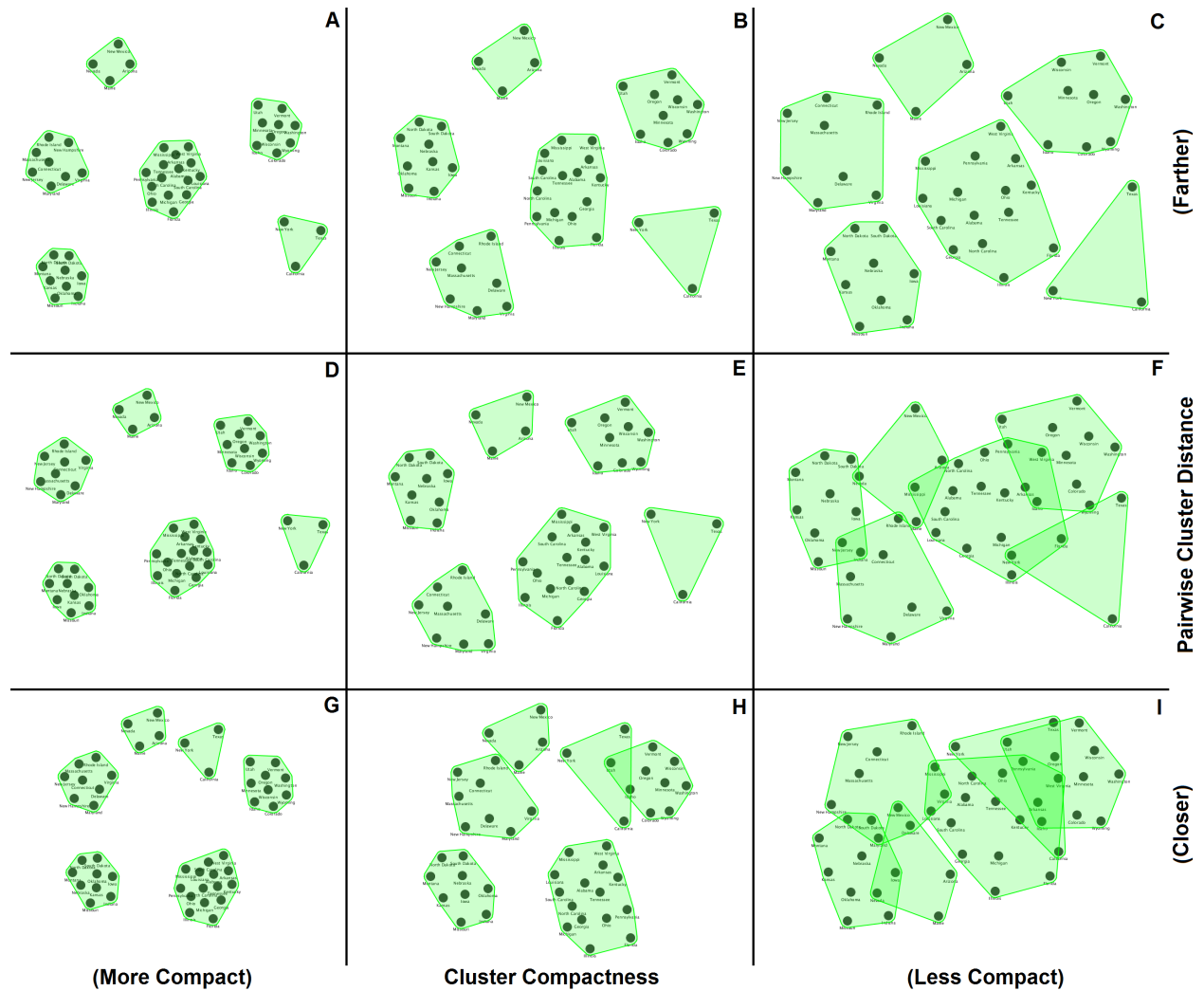


Figure 7.6: Nine views of the Census dataset with various CC , CN_I , and NN_I weights. Cluster compactness varies across the x-axis via manipulation of the CN_I and NN_I edge class weights, while pairwise cluster distance varies in the y-axis via manipulation of the CC edge class weight.

7.2.3 The Role of Edge Class Weights

In addition to selecting edge classes, each class is paired with an associated edge class weight w_e . The purpose of this weight is to influence the layout of the projection, allowing for Pollux to generate a variety of visual representations. Determining how to best lay out observations after performing cluster assignments is a dataset- and user-driven process, with the ideal

layout of the data being influenced by the insight that the analyst wishes to communicate with the visualization. There exists a natural tradeoff that is implied through manipulating the layout between the distortion of the space and the best representation of these insights. Figure 7.6 shows several Pollux layouts for the Census dataset with varied of CC , CN_I , and NN_I edge class weight values. From left to right, the CN_I and NN_I weights are altered to change the compactness of the clusters. From top to bottom, the CC weight is altered to change the pairwise distances between the clusters. As clusters become less compact, they begin to overlap, causing some ambiguity in the cluster membership assignment of some nodes (e.g., Arkansas and Ohio in Figure 7.6I). This effect is magnified as the relative pairwise distance between clusters is reduced.

7.2.4 Alternate Visual Representations

A limitation to the Pollux technique is the inherent spatial distortion required by the cluster-first projections. In other words, the Dimension Reduction Model in the Pollux pipeline assumes that cluster membership information from the Clustering Model is the primary layout factor, with weights secondary and high-dimensional distances tertiary factors. As a result, there is no way to avoid such spatial distortions without transforming the pipeline.

For example, the left panel of Figure 7.7 depicts an alternative layout of the Animals dataset. In this layout, both the projection and the layout are computed from the high-dimensional data separately. The projection then accurately reflects all pairwise distances between observations, and the cluster membership is encoded by color rather than in convex hulls. However, this is not an instance of the Pollux pipeline; rather, it matches the “Independent Algorithms” pipeline identified in our taxonomy in previous work [320]. The clusters are also not as compact and easily identifiable in this view. A Pollux representation of this Animals

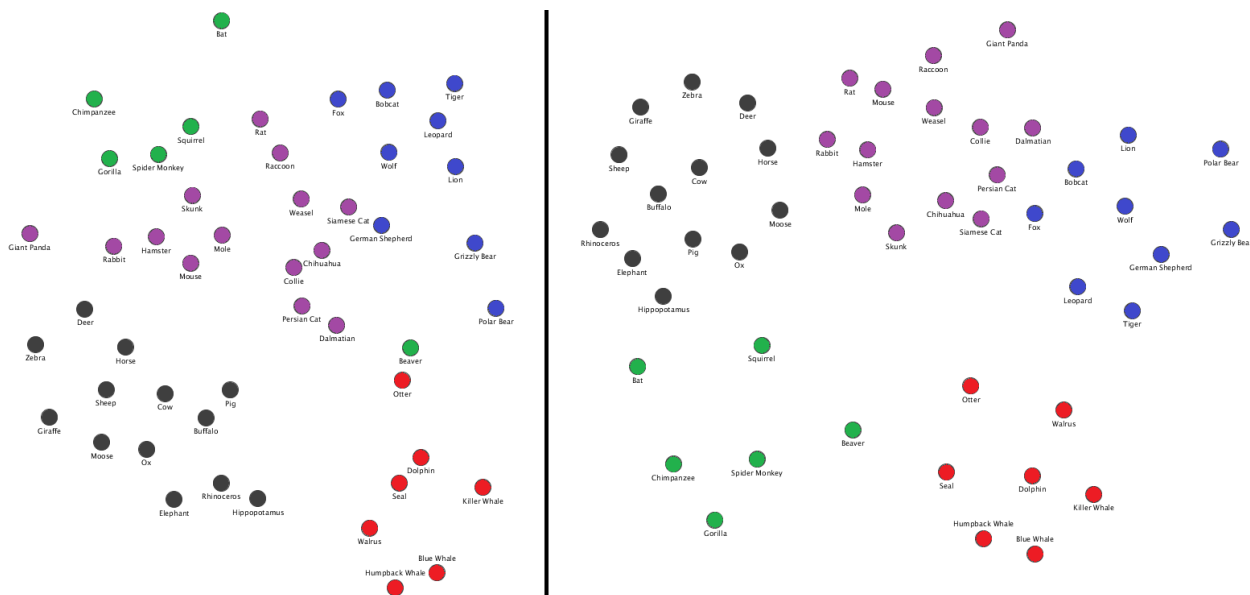


Figure 7.7: **(left)** A direct two-dimensional projection of the high-dimensional Animals data with cluster information encoded by color. **(right)** The same data in Pollux, using color encoding for clusters rather than convex hulls.

dataset using the same color mapping is provided in the right panel of Figure 7.7. In this view, the clusters are more compact and uniformly-shaped, though the view without convex hulls may not clearly imply that there is no inter-cluster similarity at the node level in this view, as the CN_E and NN_E edges were not included when generating this projection.

7.2.5 Analyst Control of Cluster Count

The Pollux examples provided thus far use the system-determined value of k to categorize and lay out the observations. However, the analyst is afforded with the ability to manipulate the value of k to update the cluster membership assignments. For example, the Fisher’s Iris dataset consists of three different species of iris: Setosa, Virginica, and Versicolor; however, Pollux only determines that two clusters exist in the dataset, and does not differentiate between the Virginica and Versicolor species (Figure 7.8 left). When the analyst updates

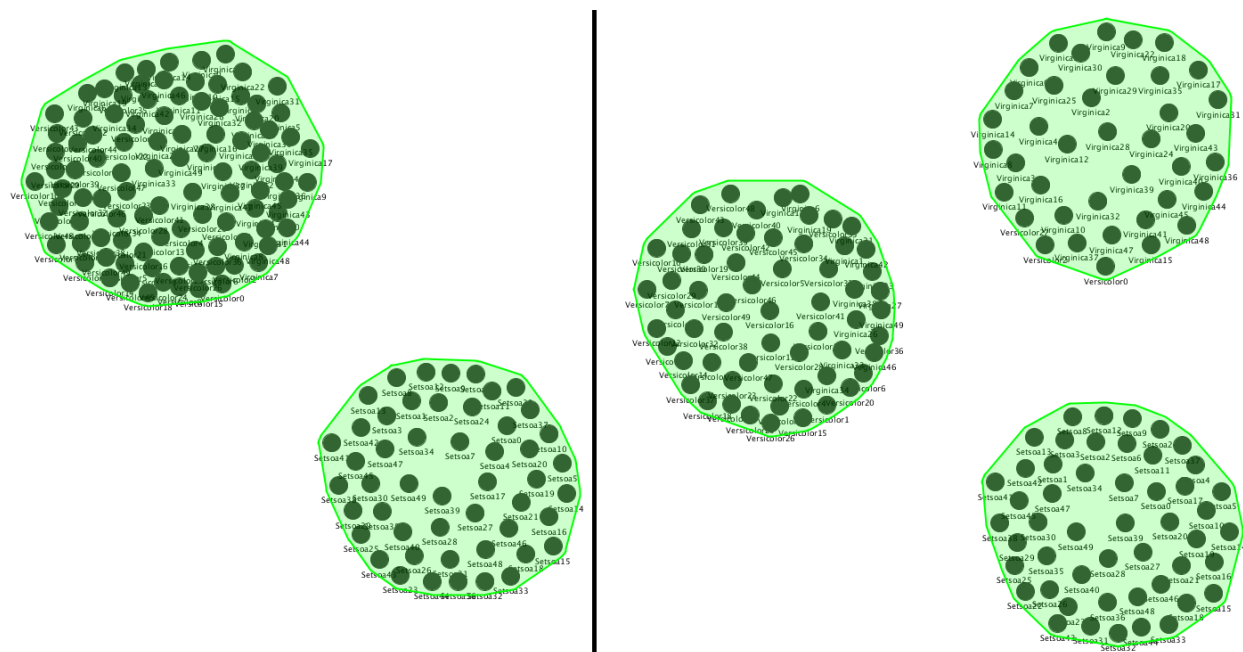


Figure 7.8: **(left)** The Fisher's Iris dataset with the system-determined two clusters. **(right)** The analyst updates the view to incorporate three clusters.

the system to force three clusters, the Virginica and Versicolor species are separated, albeit imperfectly. The analyst can then begin to perform reclassification interactions to train the system to distinguish between Virginica and Versicolor.

7.2.6 Extending the Hierarchy

Pollux as described thus far only consists of a single set of clusters containing nodes; however, the technique can be extended to include cluster hierarchies. Additional research is necessary to design interaction techniques for disambiguating between cluster reassignments in such a hierarchy. However, the benefit is the ability to visualize much larger datasets by also interactively expanding and contracting clusters. A similar technique was used by ASK-GraphView [1] to visualize hierarchically-clustered datasets several orders of magnitude larger than those demonstrated with Pollux in this work.

7.2.7 When to Learn?

As described in Section 7.1.2, Pollux contains two learning phases, identifying the reasoning behind the action of an analyst both removing an observation from a cluster as well as inserting the observation into a different cluster. However, there may be cases when the source cluster (an associated analyst intent might be “The Fox should be in the Predators cluster”) or the target cluster is not meaningful (“Alaska does not belong in the cluster of high-population states”). In these cases, only one learning phases is relevant to the interaction, and thus the second learning phase captures a portion of the interaction that has no associated analyst intent. The attribute weights are therefore updated needlessly. A second component to the interaction could help to determine which portions of the interactions have meaning. For example, a double-click interaction before dragging could indicate that the removal from the source cluster is meaningful, while holding the mouse button down for a short period before releasing could indicate that insertion into the target cluster in meaningful. Additional research is necessary to design the best interaction technique.

Disambiguating cases where only the source cluster is important, only the target cluster is important, and cases where both are important is related to the “With Respect to What” problem detailed by Self et al [273]. The original definition of this problem was focused on disambiguation of intent between interactions relationships, but the same issue is present in Pollux, albeit with fewer possible interpretations of an analyst interaction. Thus, the introduction of clusters simplifies but does not solve “With Respect to What.”

7.2.8 Multiple Distance Functions and Weight Vectors

Our implementation of Pollux uses a single shared distance function and weight vector for the Dimension Reduction and Clustering Models. However, implementations could certainly

be produced that learn separate weight vectors for each model, each of which could then use a difference distance function in processing the dataset. For example, the Dimension Reduction Model, using Manhattan distance to boost computational efficiency, might use a different weight vector than the Clustering Model, which still uses Euclidean distance to accurately determine distances between clusters in the high-dimensional data.

7.3 Evaluation

In this section, we evaluate Pollux via a usage scenario, performing reclassification interactions on the Census dataset to create a particular cluster of states, and evaluate the attribute weights learned within the system to create such an overall clustering and layout. After normalizing this dataset, we create an initial clustered projection (right panel of Figure 7.1) in which each of the 35 dimensions begins with a weight of 1. Processing this dataset with the k -means algorithm produces six clusters.

The Census Bureau defines the Midwest Region as a collection of 12 states, ranging from Ohio in the east to the Dakotas in the west [304]. In the initial projection, the cluster annotated with “Midwestern States” on the left side of Figure 7.9A already incorporates 7 of these 12 states (as well as two extra states). In order to create a Midwest Region cluster, we perform the reclassification interactions listed in the following paragraphs. The learning routine executes after each of the interactions is performed, learning new weights to reflect what has been learned about the intent of the analyst thus far.

The first interaction reclassifies Ohio as a Midwestern state, dragging it from the central center into the Midwestern States cluster (Figure 7.9A). The dimension weights are updated to reflect both the departure of Ohio from its source cluster and its introduction into the target cluster. Following the weight updates, no other states have received new cluster

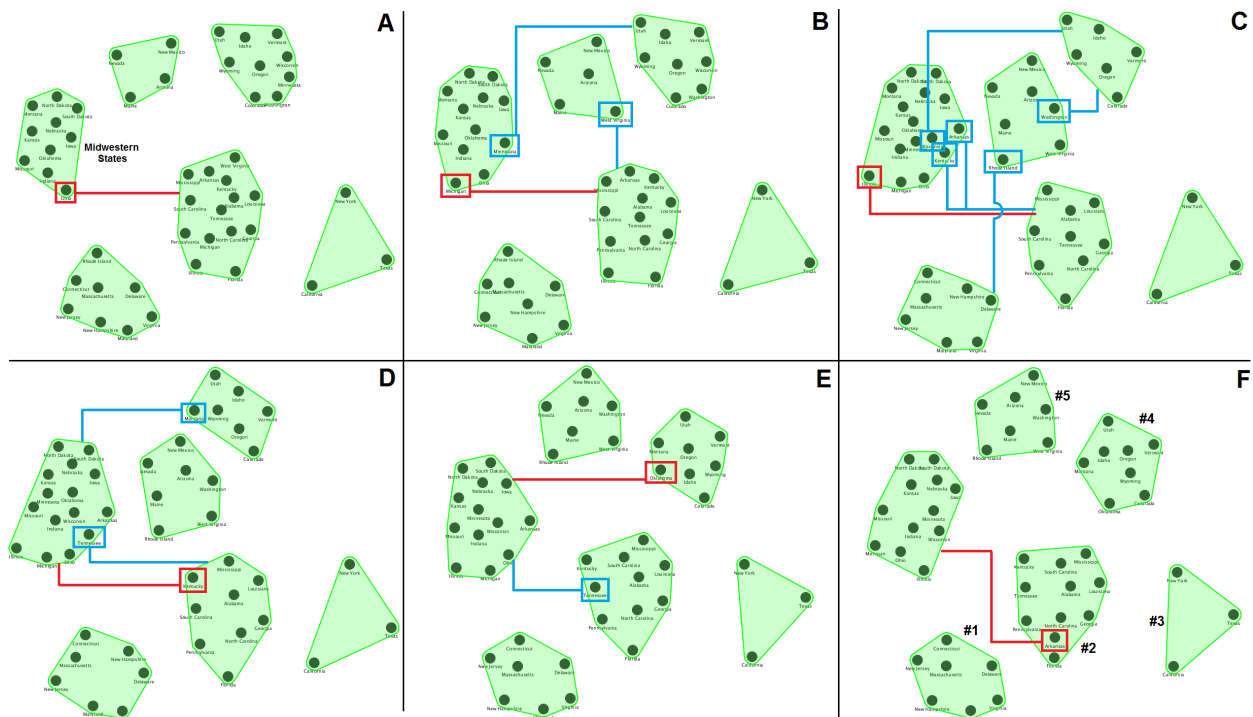


Figure 7.9: Each of the six interactions performed by the analyst in the usage scenario. Nodes enclosed by red rectangles denote analyst-driven classification updates, while nodes enclosed by blue rectangles denote classification updates made by the system in response to newly-learned weights. Lines are drawn to show observation paths from source to destination cluster.

assignments. However, the area of the Midwestern States expands slightly, both because the number of nodes has increased and because Ohio is pushed to the outskirts of the cluster. The second occurs because the system has only recorded a single interaction; it has not yet learned enough to understand the optimal position of Ohio within the cluster.

The second interaction is similar, reclassifying Michigan as a Midwestern state by transferring it from the central cluster into the Midwestern States cluster (Figure 7.9B). Following the weight updates, several updates are now apparent in both the clustering assignments and in the overall projection. Minnesota was pulled into the forming Midwestern States cluster from the upper-right. Additionally, West Virginia was reclassified as a member of the top-center cluster, departing the central cluster. Further, the three upper clusters begin

moving closer together as a result of ethnicity weights exerting more influence upon the overall graph. The states in these three clusters all have similar ethnic breakdowns, causing this effect.

The third interaction causes substantially more updates. Reclassifying Illinois as a Midwestern state also brings in Wisconsin (intended), as well as Arkansas and Kentucky (unintended) (Figure 7.9C). As a result, there are now four states which must be removed from the cluster, but all five new states have now been introduced. In addition to these updates to the Midwestern States cluster, the states of Rhode Island and Washington were transferred into the upper-center cluster. The three upper clusters continue their drift from the previous interaction.

The fourth interaction begins the removal of the unwanted states, and also demonstrates the inertia of the learning routine. Removing Kentucky from the Midwestern States cluster and positioning it into the cluster which appears most sensible based on geography (the center cluster) also results in the automatic removal of Montana (hoped for), but it additionally brings unwanted Tennessee into the Midwestern States cluster (Figure 7.9D). Tennessee and Kentucky are quite similar states, and Tennessee was close to being relocated into the Midwestern States cluster before this interaction. Following the removal of Kentucky, the dimension weights just enough to finally pull Tennessee in. The upper-center and upper-right clusters were temporarily overlapping after this interaction, though they eventually separated as the projection stabilized.

Finally, the fifth and sixth interactions had minimal impact beyond the removal of states from the Midwestern States cluster. Reclassifying Oklahoma into the upper-right cluster removed Tennessee and returned it to the central cluster, where it was classified prior to the fourth interaction (Figure 7.9E). Reclassifying Arkansas into the central cluster had no other cluster assignment effects (Figure 7.9F).

The result of this set of six interactions is the formation of a cluster of the 12 Midwestern states, as well as five other clusters which saw occasional updates based upon the analyst's interactions with the Midwestern States cluster. Progressing counterclockwise (with the clusters labeled #1–5 in Figure 7.9F), these clusters could be mapped with semantic meanings such as Northeastern States (#1), East Coast States (#2), High-Population States (#3), Low-Population States (#4), and a cluster that is difficult to label, but contains states that lean towards elderly, rural populations with lower than average per capita income (#5).

Figure 7.10 shows a selection of six attribute weights, their value updates following each analyst interaction, and their influence on the overall projection as a result. The weight with consistently the most influence over the projection, the percentage of residents with a high school diploma, matches well with the states that the analyst reclassified: Ohio, Michigan, and Illinois each rate between #21 and #26 when the states are sorted by this attribute, while Kentucky, Oklahoma, and Arkansas are #2, #15, and #4 respectively. Indeed, the only time when the high school graduation rate decreases in influence was after Oklahoma was reclassified. The attribute weights associated with Caucasian, elderly, and male residents consistently declined through the interaction sequence, while the attribute weight for Hispanic residents varied in influence by that group's population in each state. The 1990 population was selected in the figure to demonstrate a dimension that seemed to have no meaningful impact on the overall projection, with the value of this attribute weight oscillating about the default of 1 as the interaction sequence progressed.

7.4 Discussion

The overarching focus of this research direction is to continue to explore the complex interplay between dimension reduction and clustering algorithms in both systems and in humans.

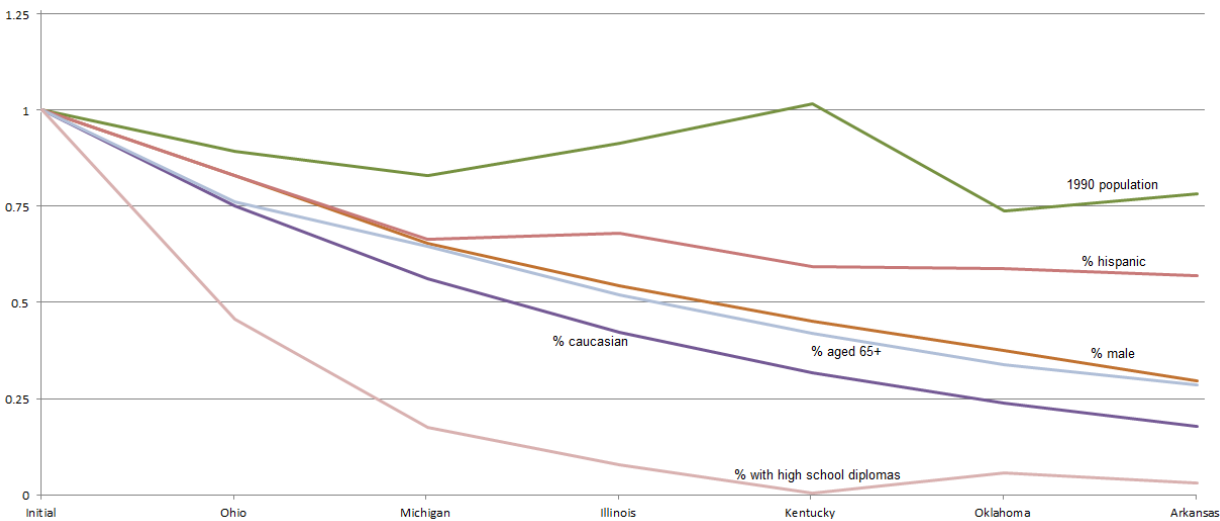


Figure 7.10: A selection of six attribute weights and their respective value updates during the six analyst interactions.

As noted in the introduction, these algorithms serve different cognitive purposes but can naturally coexist within a projection of high-dimensional data. Understanding how analysts interpret and interact with such visualizations is a long-term goal, of which Pollux represents one point in the overall design space. That said, the variety of visual representations that can be produced by Pollux through modifications to the edge class selection and weights, cluster representation, distance function selection, and learning method demonstrate the flexibility of this framework for visualizing high-dimensional datasets.

Tools like Pollux and Castor can be supplemented by additional views that convey additional information about the visualization. For example, the weight graph from Figure 7.10 provides additional context beyond the list of weights presented in the tool. Similarly, dynamic graphs of projection stress and cluster quality such as those provided in the next chapter (Figures 8.5, 8.6, and 8.7) can provide feedback to analysts regarding the quality of their visualization after each interaction. Such Explainable AI techniques provide the analyst with additional insight beyond the projection and clustering assignments themselves.

7.4.1 Limitations and Future Work

One notable limitation of the work presented here is the scale of data tested. Though we did vary the shape of the datasets visualized (e.g., many more observations than dimensions, similar numbers of observations and dimensions), the true power of this cluster-first technique lies in its scalability. The current k -means and force-directed implementation of Pollux was the limiting factor in experimenting with larger data scale, and so we chose to focus this work on demonstrating the use and success of the reclassification, learning, and layout technique. Our next step in development is to re-implement the system with scalability in mind. An alternative means of supporting larger datasets would be to initialize the projection and clustering by strategically sampling from the overall dataset, and then foraging additional data as needed in response to the interactions performed by the analyst. Indeed, this foraging behavior has been demonstrated in similar systems for text data [38, 39, 99], but is not often found in systems intended for quantitative data.

Further, we have not yet performed a user study to test the usability of this cluster-first technique in contrast to existing layout-first techniques. Our demonstration of Pollux in this work is limited to a usage scenario demonstration of the technique. A full study to examine the similarities and differences in insights generated by each technique is currently planned.

Finally, a limitation of the Pollux technique is the distortion of space to create compact clusters, a distortion that goes beyond that which is already necessary when projecting into a low-dimensional space. Beyond examining the underlying model weights, Pollux currently lacks a method for demonstrating to analysts whether or not their current clustering is meaningful. In other words, is the clustering that was constructed by the analyst supported by the data, or does it force nonsensical constraints upon the data in order to generate the current clustering? There are a variety of methods that we are considering to visualize

the quality of clusters and to quantify the fitness of user-imposed constraints, with enough options under discussion to necessitate a further study, taking this issue beyond the scope of this work.

7.5 Conclusion

This work presents Pollux, a system that combines clustering and dimension reduction algorithms in a cluster-first framework to efficiently produce an interactive visualization of an input dataset. By interacting with the visualization, an analyst provides feedback to the underlying models, incrementally training the models to produce representations that reflect the current exploration interests of the analyst. We discuss means by which the default implementation of Pollux can be altered or extended, and we demonstrate the effectiveness of the interactive reclassification interaction via a usage scenario. The flexibility demonstrated by Pollux presents an interesting tool to continue to develop, with several future studies planned.

Chapter 8

Analyzing Pipeline Order Via Case Studies

This dissertation has thus far explored the design space for cooperative dimension reduction and clustering algorithms within interactive visual analytics tools. In particular, Chapter 4 identifies six pipelines for methods to combine these algorithms, altering the sequence of these algorithms and the ways by which they interact while transforming a dataset into a visualization. Chapters 6 and 7 implement two of these pipelines, the Dimension Reduction Preprocessing for Clustering (Castor) and the Clustering Preprocessing for Dimension Reduction (Pollux) versions. We argue that the order of those dimension reduction and clustering computational models is important, as changing the model order and interactions will change both the initial state of the visualization and the exploration path of the analyst. Our goal in this work is to evaluate the two prototype tools the correspond to those pipelines, as well as introducing a third tool (Gemini) as a de facto control tool. In particular, we note the following contributions:

1. The evaluation of these tools, both by an insight-driven usage scenario and through a quantitative tradeoff evaluation.
2. Design considerations for future developers and avenues of future research in this dimension reduction and clustering space.

We developed and will demonstrate each of these tools through use of the same datasets. These datasets are summarized in the last chapter in Table 7.1. Two of these datasets (Animals and States) had approximately the same number of observations and dimensions, while the third dataset (Fisher’s Iris) has substantially more observations than dimensions. The Animals dataset was the primary dataset used for development and early prototype demonstrations during our iterative design process. Each of these datasets was normalized so that attribute values ranged from zero to 100 before being processed and visualized.

Each of these datasets is relatively small, particularly when considering the enormous size of datasets processed by current machine learning techniques. Our goal with this work is to explore tradeoffs in the computational pipelines, rather than to demonstrate the scalability of our technique. Through development testing on an 8 year old desktop machine (i7 Sandy Bridge processor, 16GB RAM), we found that animations were no longer smooth after approximately 500 observations were included in the projection, and interactions were difficult to justify as “real-time” shortly after that. Further, we focus on quantitative datasets in this analysis rather than collections of text documents, as techniques for computing similarities of text require a conversion to numerical data.

We selected these three pipelines for implementation from the six in Figure 4.3 because we felt they were both the more straightforward and more interesting cases to compare the effects of algorithm order. The other three pipeline options involved more intricate interactions between the dimension reduction and clustering behavior, including global/local distinctions and iterating computations across the two algorithms. By looking at these three, we felt that we could draw the best conclusions about the influence of algorithm order.

Here, we evaluate our tools by two separate methods. In Sect. 8.2, we take an insight-based approach to show how the analyst can reach a range of conclusions about three datasets in each of the tools. In Sect. 8.3, we take a quantitative approach, examining the tradeoff

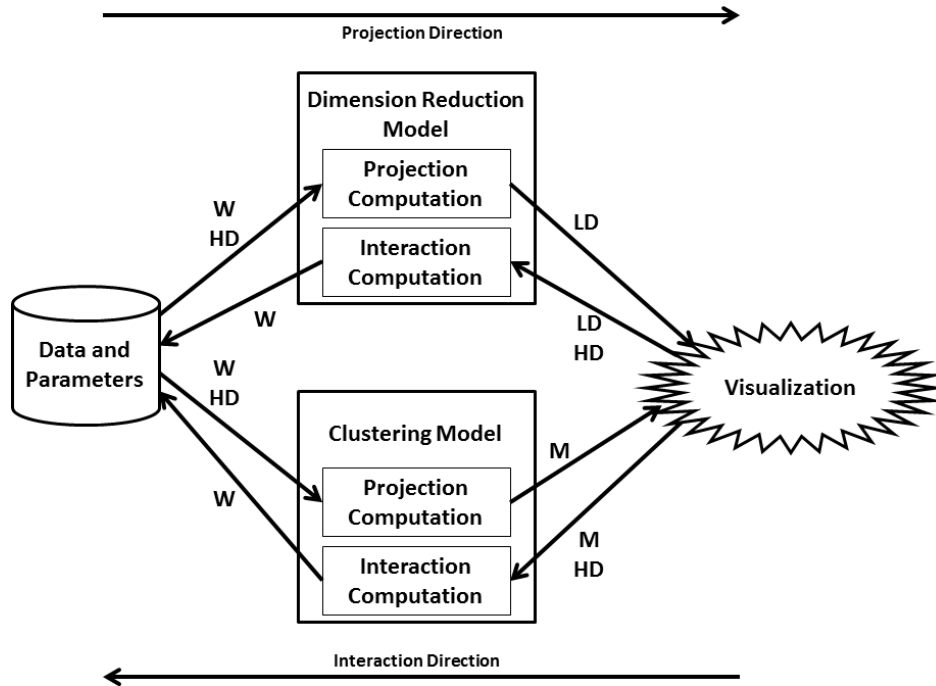


Figure 8.1: Gemini pipeline: both the dimension reduction and clustering algorithms are processed on the high-dimensional data in parallel.

between these pipelines by measuring both the stress within the projection and intra-cluster distances within the cluster membership assignments.

8.1 Gemini

Gemini executes both the dimension reduction and clustering processes in parallel, using the high-dimensional data as input for both algorithms. The goal of such a pipeline is to show the most accurate representation of the output of each algorithm, without one algorithm influencing the behavior of the other in generating the visualization. Figure 8.1 displays a component-level pipeline of the system. The dimension reduction and clustering algorithms are stored respectively in Dimension Reduction and Clustering Models. In each model is a Projection Computation and an Interaction Computation. As indicated in this figure, arrows

flowing from left-to-right are handling the projection of the data, while arrows flowing from right-to-left are responses to analyst interactions. Labels on each of the arrows connecting the pipeline components signify what system data is the input and output of these processes (where W refers to the vector of dimension weights, HD is the high-dimensional data, LD is the low-dimensional data, and M are cluster membership assignments).

Considering the Dimension Reduction Model first, the current dimension weights and the high-dimensional data are passed into the Projection Computation, which generates the low-dimensional coordinates used for the visualization. When an analyst drags a node in the projection without the drag interaction crossing the cluster boundary, the Interaction Computation of the Dimension Reduction Model responds, using both the high- and low-dimensional data to understand the effects of the interaction through the learning behavior described in the last section, generating a new set of weights. These new weights are then applied immediately to the entire projection, yielding a new layout by executing the Projection Computation of both Models again.

The Clustering Model exhibits similar behavior. Again, the current dimension weights and the high-dimensional data are passed into the Projection Computation, clustering the data through the weighted k -means algorithm and generating cluster membership assignments, which are then rendered in the projection. When an analyst drags a node across a cluster boundary, the Interaction Computation will respond, using both the high-dimensional data and cluster membership changes to understand the effects of the interaction, and again generating a new set of weights. This new set of weights is again applied to the entire projection by executing the Projection Computation of both Models (Figure 8.2).

An example of running this pipeline on the animals dataset [195] is shown in the leftmost panel of Figure 8.3. An immediately-obvious feature of this visualization is the overlap between two of the clusters as the result of the positioning of the Beaver in the projection

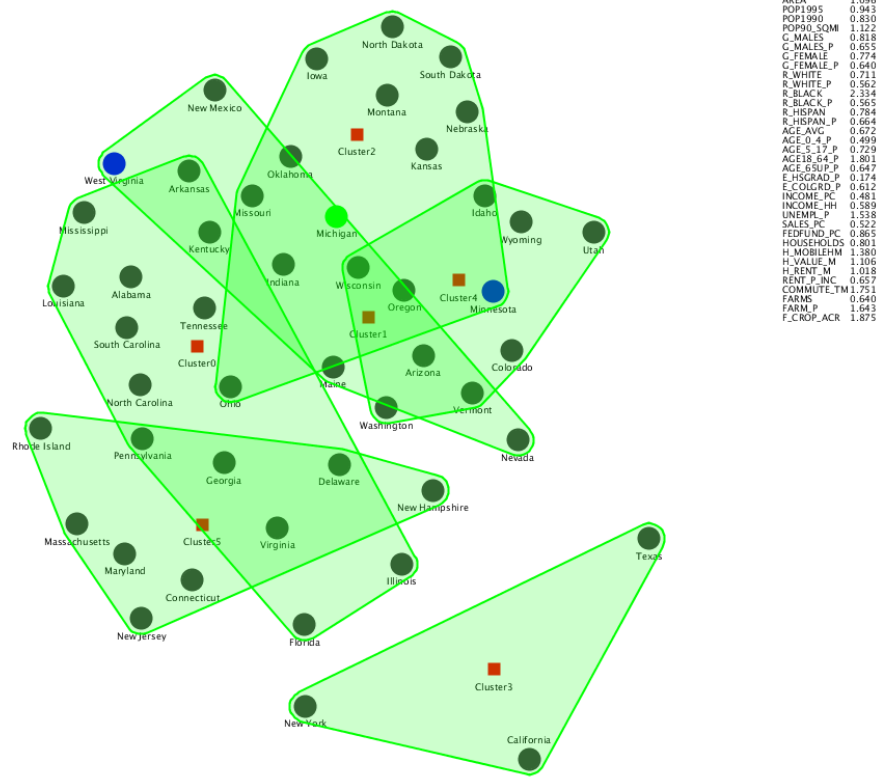


Figure 8.2: After Michigan was assigned to Cluster 2 (green node), both West Virginia and Minnesota were reclassified in response (blue nodes).

from the Dimension Reduction Model. The amount of overdraw is substantial enough to make the cluster assignments of several observations (Spider Monkey, Rat, Raccoon, Weasel, and Chihuahua) ambiguous.

8.2 Insight Usage Scenario

Here, we discuss and categorize insights identified from each of our three tools when visualizing each of the three datasets, understanding which facets of the data an analyst may glean from inspection of a visualization [233, 269].

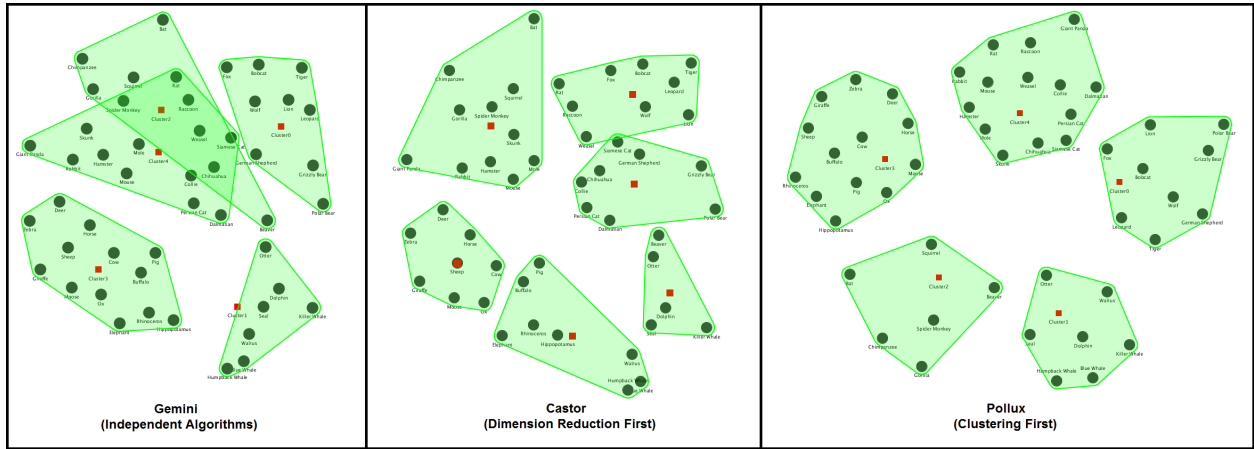


Figure 8.3: The animals dataset [195] loaded into Gemini (left), Castor (center), and Pollux (right). Note that the layout is nearly identical in Gemini and Castor, while the clustering assignments are the same in Gemini and Pollux.

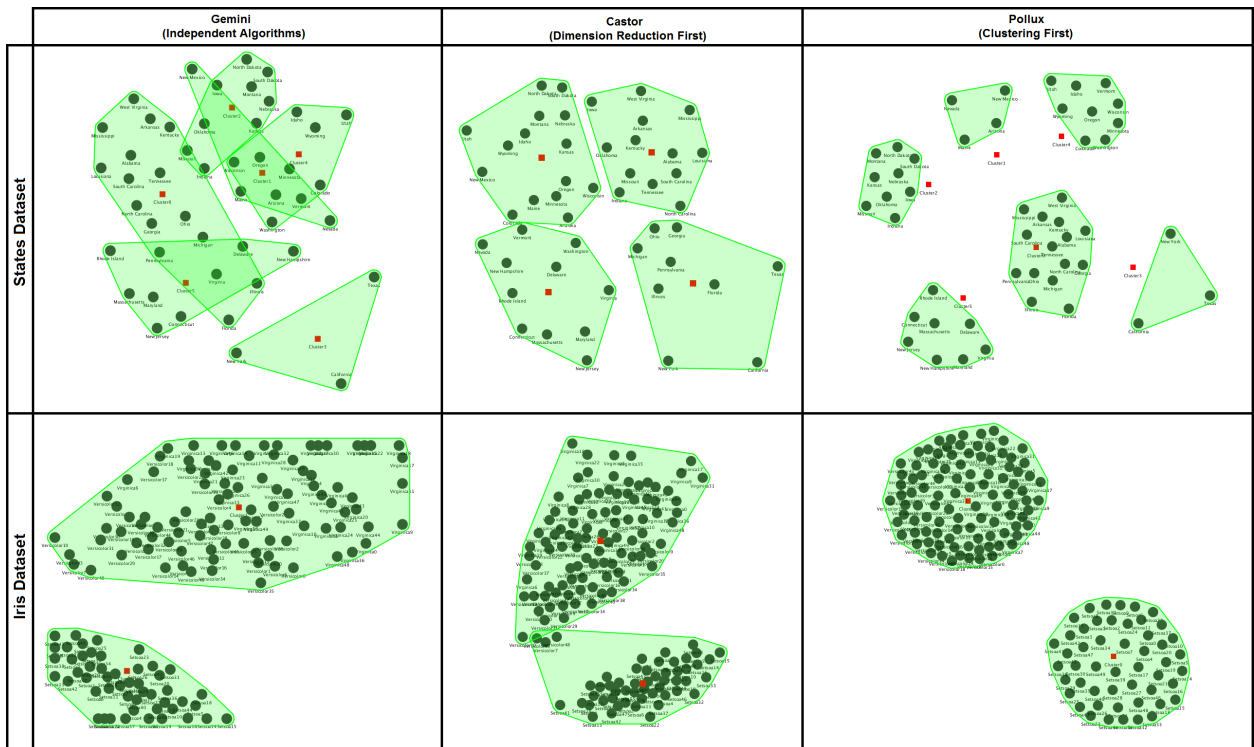


Figure 8.4: The initial view of the states dataset (top row) and Fisher's Iris dataset (bottom row) visualized in all three tools.

8.2.1 Gemini

The overlapping clusters in the Animals and States datasets were a distraction from being able to quickly gain insight on the data. We can see that there is some obvious semantic meaning to some of the groups of animals; for example, there is an aquatic animals group in the lower-right and a group of large grazing animals in the lower-left. Other animal groups appear to be more of a mix, such as the group that combines the Beaver, the Bat, and the Gorilla. The same is true for the States dataset. Some groups have obvious semantic meaning, such as the high-population states in the bottom-right and the northeastern states in one of the overlapped clusters. But then again, Maine is grouped with several states in the desert southwest. In the Iris dataset, it is obvious from the node labels that the Versicolor and Virginica nodes should have been separated into separate clusters, but we can conclude from seeing this projection that those two classes are quite similar to each other and different from the Setosa class. Further, there does appear to be a distinct boundary within the large cluster, with most of the Versicolor on the left and most of the Virginica on the right, informing us that they are indeed distinct groups.

8.2.2 Castor

In contrast to the insights that we identified in Gemini, it is more difficult to find semantic meaning in the clusters in Castor. With the Animals dataset, there appears to be a partial aquatic animals cluster that includes the Beaver this time, but neglects the Walrus and the two whales. There also appears to be a pets cluster that also includes the Grizzly Bear and Polar Bear, which are certainly not pets. The same is true in the States dataset – there appears to be no straightforward way to explain any of the clusters geographically, as there are always exceptions included. The bottom-right cluster again contains many high-

population states, but also misses several. The clusters are more distinct and meaningful with the Iris dataset, and ignoring the issue with k -means not clustering optimally, this is a useful view, though functionally identical to the view provided by Gemini.

If we ignore the cluster boundaries, the overall spatialization reveals trends in the datasets. The carnivorous animals are positioned towards the upper-right of the projection, the population of the states generally increases from the top of the projection to the bottom, and we can see the two groups of flowers with an implicit border between the Versicolor and Virginica classes in the Iris dataset. Again, these are the same conclusions that we would draw if we removed the clusters from Gemini and only used the spatialization in that tool.

8.2.3 Pollux

Using Pollux, the groupings immediately appear evident within the projections. As these are the same groupings that were computed in Gemini, the insights that we gain are the same. However, the fact that all clusters are visibly separated with no overlap makes these insights much easier to detect. Further, Pollux often allows us to see relationships between the groups more easily. As we noted in Castor, we could see a trend in which the carnivorous animals were aligned towards one corner of the projection. Similarly, we see a group of carnivorous animals to the right, though still missing a few of the meat-eating cats and dogs in the cluster immediately beside it. Similarly, the aquatic animals cluster is again missing the Beaver, but the Beaver is also in a nearby cluster. The groups of states from Gemini are still evident in Pollux, though the relationships between them are a bit more clear. For example, at the top are two clusters that combine to collect most of the western states (though still including Maine). The groupings in the Iris dataset also remain the same, though we note that with this dataset, the lack of the spatialization makes it difficult to identify the implicit

boundary between the Versicolor and Virginica flowers, and further makes it impossible to judge the magnitude of the difference between the similar Versicolor and Virginica groups and the different Setosa group.

8.2.4 Insight Trends

The insights provided above touch on several common themes. In some cases, keeping both the clustering and the spatialization is quite effective at communicating more information (e.g., Gemini Iris vs. Pollux Iris) than if the spatialization is removed. However, this is not true in other cases, in which there are significant overdraw issues which could be overcome by removing the spatialization (e.g., Gemini States vs. Pollux States). The value of keeping the spatialization appears to be dataset-dependent, and also potentially dependent on the number of clusters found in the data. The second explanation also covers the reason for relationships between clusters in Pollux being apparent in some of the datasets but not others – it is more difficult to judge the magnitude of a difference between two clusters than it is when there are six clusters to compare in pairwise fashion. Overall, it is difficult to justify the inclusion of clustering with Castor, with the exception of the explanation provided in Sect. 8.1 that performing dimension reduction first improves clustering efficiency. Finally, we note that many of our insight observations were about groups, which may be influenced by the cluster encoding. We discuss this further in Limitations in the next section.

8.3 Quantitative Evaluation

For our quantitative evaluation, we measure both the stress within the projection and the intra-cluster similarity of the clustering, seeking to better understand the tradeoff in com-

binning these algorithms in the same projection. In other words, how does the dimension reduction-first approach affect stress differently than cluster-first, and similarly what affects do these pipelines have on intra-cluster distance? We first mathematically define these two quality metrics in Sect. 8.3.1. Following this, we examine these metrics in the initial states of each of the three datasets in Sect. 8.3.2. Finally, we trace these metrics during a set of interactions in each of the three tools, identifying how they change as an analyst interacts in Sect. 8.3.3.

8.3.1 Quality Metric Definitions

Dimension Reduction Stress: As stated in Chapter 2, the goal of dimension reduction algorithms is to find low-dimensional coordinates that preserve the relationships that exist in the high-dimensional space. When relationships are measured by distance, as they are in our three tools, the preservation is accomplished by minimizing a stress function [191] similar to that in Eqn. 8.1.

$$stress = \min_{r_{1..n}} \sum_{1 \leq i < j \leq N} (\delta_l(n_i, n_j) - \delta_h(w, n_i, n_j))^2 \quad (8.1)$$

In this equation, we seek to minimize *stress* while solving for coordinates $r_{1..n}$. For observations n_i and n_j in a dataset of size N , $\delta_l(n_i, n_j)$ and $d_h(w, n_i, n_j)$ represent a distance metric in the low-dimensional space and a weighted distance metric in the high-dimensional space, respectively. Typically the low-dimensional metric is Euclidean distance, and in our tools, the high-dimensional metric is a weighted Euclidean distance δ_w such that $\delta_w(n_i, n_j) = \sqrt{\sum_{k=1}^p w_k (x_{ik} - x_{jk})^2}$. Important to note in this computation is that we only measure stress between pairs of nodes that are connected by an edge, which is relevant to our hierarchical implementation of cluster-first Pollux.

Intra-cluster Distance: In contrast, the goal of k -means is to minimize an intra-cluster

distance measure (*icdist* in Eqn. 8.2), trying to determine the number and set of clusters with the smallest difference in the properties of the nodes within each cluster. Of course, the optimal solution is to give each node its own cluster, leading to a summed *icdist* distance of zero. Therefore, the optimal number of clusters is determined by iteratively adding another cluster, computing *k*-means until it converges to a solution, measuring *icdist* at each cluster cardinality, and finding an “elbow” in that graph where the marginal gain of adding another cluster does not justify the addition of that cluster. A common method is to measure this by a sum of squared errors, as shown in Eqn. 8.2.

$$icdist = \min \sum_{k=1}^K \sum_{n \in k} \sqrt{\delta(n - c_k)^2} \quad (8.2)$$

Here, we square the distance between each observation *n* and its cluster centroid *c_k*, take the square root of each of these values to make the sum a more tractable number, and then sum these distances across each node-centroid pair in all *K* clusters. An alternative measurement is to perform this intra-cluster measure with the sum of all pairs of nodes within a cluster. The result of this computation will be approximately the same. Note that this computation is performed in the high-dimensional space in all three tools. Finally, we include an analysis using the Davies-Bouldin Index from Chapter 3, with the formula provided in Eqn. 3.2.

8.3.2 Quantitative Measures on Initial States

Figure 8.5 displays the overall projection stress using all three tools on each of our test datasets. An immediate observation when inspecting this chart is that the stress in the Pollux projection is quite similar to the other tools for the Iris dataset but is much smaller in the other two datasets. This is due to the fact that each of the tools created two nearly-equal clusters (similar in cluster assignment but different in shape). Because of the size

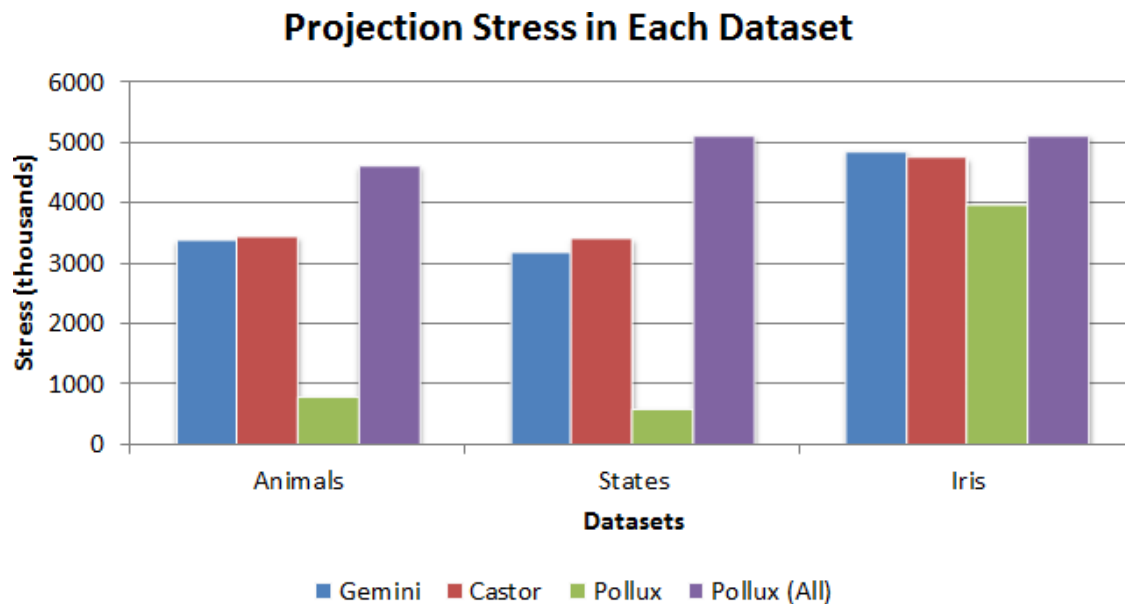


Figure 8.5: The projection stress of each dataset while using each tool.

of the dataset, the clusters were not as compact. In contrast, the stress of Pollux when projecting the Animals and States datasets is far smaller, as a result of the compactness of each of the clusters in Pollux when compared to those of Gemini and Castor. We note that this stress measure is artificially small, since we are only considering edges that are present in the graph structure we have created (described in Sect. 8.2.3). The bars labeled Pollux (All) include node–centroid and/or node–node edges that cross cluster boundaries, making the overall stress in these projections considerably higher.

Further, it is worth noting that the stresses of Gemini and Castor are similar but not identical. Both tools are computing the force-directed simulation on the high-dimensional data, so this should be an expected outcome. The fact that these measures are merely similar rather than being equal is due to the nondeterministic layout of the force-directed simulation – the projections are very similar but are not identical. Overall, the stress of a projection across our three tools is dataset-dependent, with the number of clusters directly influencing the stress value.

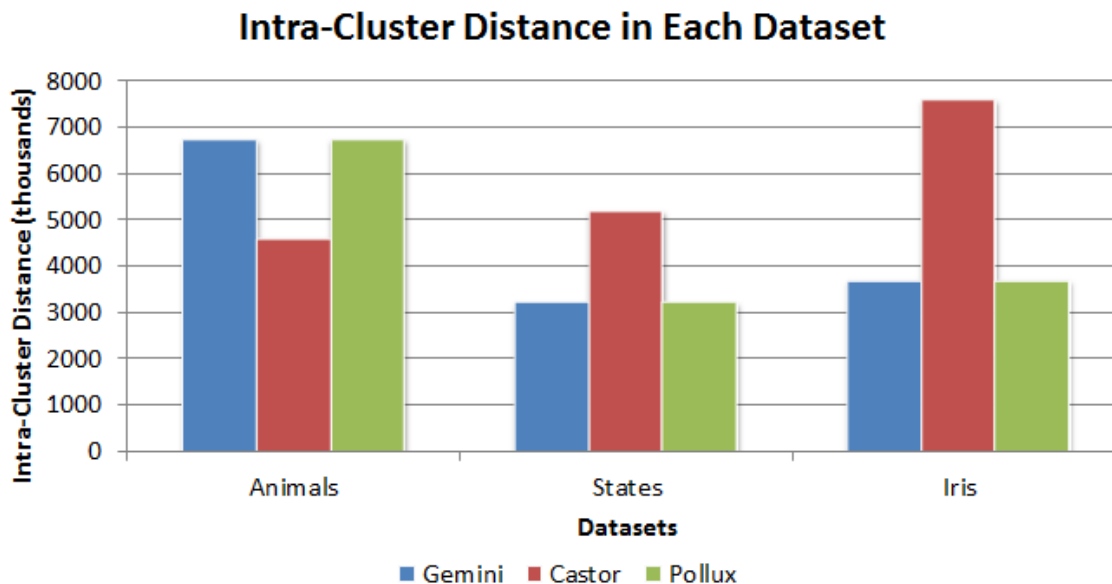


Figure 8.6: The total intra-clustering distance of each dataset while using each tool.

Similarly, Figure 8.6 displays the intra-cluster distance using all three tools on each of our test datasets. In this case, these distances are identical for Gemini and Pollux, since the cluster memberships of all observations in these tools is computed on the high-dimensional data, while the clusters of Castor are computed in the projection.

The intra-cluster distance of the Castor visualization is smaller than the others when visualizing the Animal dataset, but is larger for the States dataset and roughly double for the Iris dataset. The issue with the Iris dataset is obviously a k -means algorithm issue, as seen in the bottom-center image of Figure 8.4. A density-based clustering algorithm would not have assigned three of the Versicolor nodes to the Setosa cluster.

The cause of the difference is not so apparent in the other two datasets, but can still be deduced. The k -means algorithm determined that there are four clusters in the Castor States visualization when computed in the projection coordinates, but six clusters in the other two tools when computed in the high-dimensional space. The four clusters in Castor are less compact than the six in the other tools, leading to a larger intra-cluster distance.

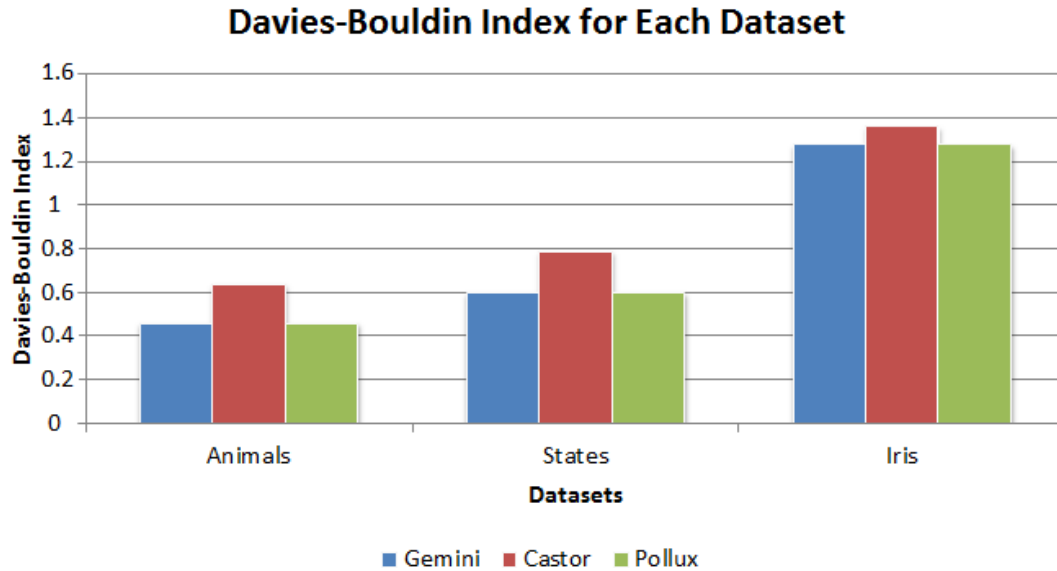


Figure 8.7: The Davies-Bouldin Index for each dataset while using each tool.

The opposite effect holds for the Animals dataset. Here, Castor contains six clusters, and opposed to five clusters in Gemini and Pollux. Therefore, the intra-cluster distance is smaller in the more compact Castor clusters. Again, we see that this intra-cluster distance quality measure is also dataset-dependent.

Finally, Figure 8.7 displays the Davies-Bouldin Index using all three tools on each of our test datasets. Because this statistic presents a ratio of the average within-cluster distances over the inter-cluster distance, smaller values represent better clustering assignments. Gemini and Pollux again have the same statistics because of the identical clustering result. In all cases, Castor scored worse than Gemini and Pollux, a result of the low-dimensional clustering. The difference was smaller with the Iris dataset due to the smaller number of large clusters.

8.3.3 Quantitative Measures Across Interactions

This evaluation repeats the Usage Scenario from Section 7.3 with some additional analysis. The United States Census Bureau defines the Midwestern region as a collection of 12 states,

ranging from Ohio in the east to the Dakotas in the west [304]. We sought to identify how the stress and intra-cluster distance change when tasked with forming this group. The results are presented in Figure 8.8, the initial starting value of each measure normalized to 1 to show the magnitude of the effect.

At the beginning, 7 of the 12 states were already grouped appropriately in a cluster; we needed to add five states to the nascent cluster and remove two others. We performed the following interactions in Pollux and Gemini:

1. Move Ohio into the cluster.
2. Move Michigan into the cluster, an action which also corresponded to the system beginning to learn our intent and pulling Minnesota in automatically.
3. Move Illinois into the cluster, which also automatically brought in Wisconsin (hoped for) as well as Arkansas and Kentucky (unintended).
4. Remove Kentucky, an action which also resulted in the automatic removal of Montana.
5. Remove Oklahoma.
6. Remove Arkansas

Following this set of six interactions, the variable that the system determined to be most applicable to our interactions by a significant margin was farms per capita, followed distantly by unemployment rate and high school education rate. We followed the same set of interactions to produce this cluster in Gemini. Because Castor had a different starting condition with four larger clusters, the task was more complex and required nine interactions.

In Gemini, as the cluster quality improved, the stress in the projection increased greatly, as a result of the spread and increasing overlap of the Midwestern Cluster in the projection. In Castor, both stress and intra-cluster quality became worse at the beginning, but then stabilized for the last several interactions. In Pollux, the stress in the projection remained

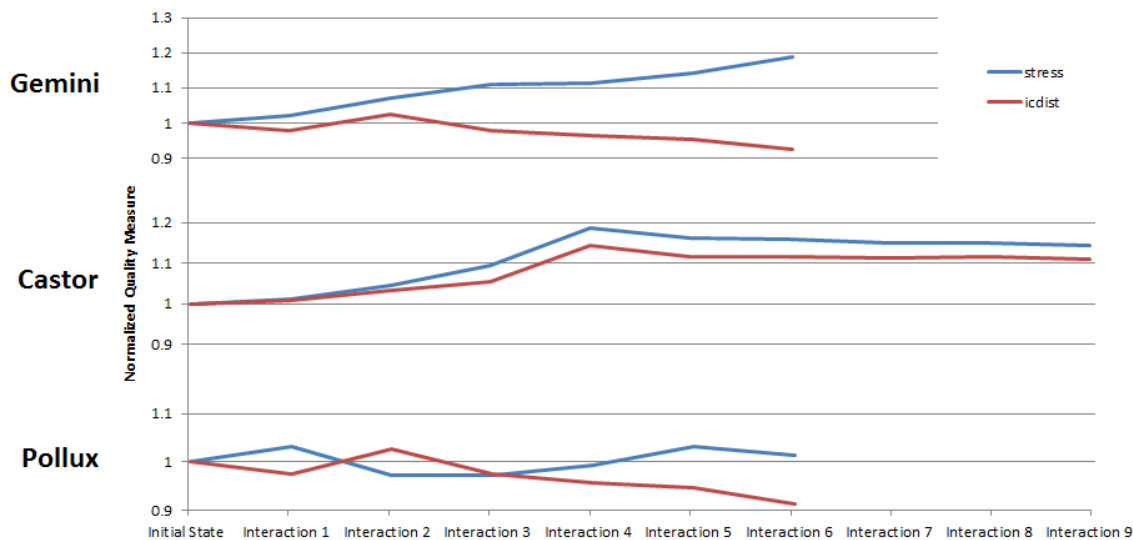


Figure 8.8: The normalized change in projection stress and intra-cluster distance over a set of interactions in each tool.

relatively constant with some fluctuations as the cluster quality improved, which can be attributed to the compactness of the clusters in this tool. By this measure over several interactions, Pollux appears to be the most stable tool.

8.4 Discussion

We begin our discussion by following up on some common themes regarding the use of our tools that were noted in our evaluation in Sect. 8.4.1, and as a result, we propose a set of design considerations to benefit future system designers in this space in Sect. 8.4.2. Finally, we discuss limitations and future work in Sect. 8.4.3.

8.4.1 Tool Quality Summary

From the evaluation performed in the last section, we note several common themes. First, clustering in the high-dimensional space appears to be far more useful than clustering in the

projection. This is true when considering the accuracy of the clustering result, as well as for deducing the semantic meaning that could be applied to the clusters. Further, Gemini and Pollux were both more stable to interact with than Castor, as a result of the clustering inaccuracy in the projection.

Second, the cluster overdraw issue in Gemini made it difficult to extract insight from the datasets projected in that tool. A different cluster encoding method would solve this challenge, though it may also lead to other confusion from users (e.g., “Why is animal X grouped in the red cluster when it is in a completely different part of the projection?”)

Third, the high-dimensional spatialization is also useful when trying to interpret distances between clusters. This was especially noticeable in the initial state of the Iris dataset, but was also apparent during interactions when trying to create the Midwestern States cluster. Depending on the dataset, this information can be easily lost in a tool like Pollux, and so a designer should carefully consider the tasks that their system intends to support when selecting a pipeline.

8.4.2 Design Considerations

We propose the following design considerations for future developers who may wish to create tools that combine these algorithm families:

1. **Decide which tasks to support.** We noted that the spatialization is better able to show relationships between observations, while the clusters help to identify common groups. If identifying these groups is most important, then a Pollux-like tool would be ideal. If identifying broader patterns is important, then incorporating the spatialization into such a system is necessary.
2. **What does the data look like?** This is difficult to answer without running a test

clustering on the dataset, but we noted that the number of clusters detected in the data and their relationships to each other can have an effect on the resulting visualization.

3. **How should clusters be encoded?** Though we have not yet experimented with altering the clustering encodings, there are certainly datasets that are not well suited to a convex hull visualization. Alternatives such as encoding cluster membership through color should also be considered.
4. **How should the dimension weights be handled?** Again, we have not yet experimented with altering the treatment of the dimension weights, but it is certainly possible that the dimension reduction and clustering algorithms could each maintain a separate weight vector. Similarly, interactions within a cluster could be kept local, resulting in a unique weight vector for each cluster.

8.4.3 Limitations and Future Work

We note several limitations of our study and implementations that we wish to correct in future work. First, our insight-driven usage scenario was only the authors drawing insight from these tools to demonstrate their characteristics. A user study involving 20–30 participants would yield a much stronger result and better conclusions about what analysts can gain from using these tools. Further, we did not explore visualization tradeoffs in these tools. For example, the participants' insights might have been different had we rendered cluster membership with nodes of various colors rather than convex hulls. We plan to perform such a study on the effect of cluster encoding on interactive clustering in the future; until then, some work in the literature has touched on the non-interactive portion of this area [168, 265].

Additionally, we limited the interactions that were permitted in our tools to a small set of what is possible, reducing the interaction space to a few representative interactions. As we

did so, we created a one-to-one mapping between interaction and intent that is fixed within the system. However, future work could incorporate a more dynamic method of learning the intent of an interaction. In other words, rather than creating a fixed rule of “interaction X causes behavior Y,” the system could learn the appropriate behavior to infer from the interaction. Such a system can rely on the interaction history recorded by the system, and could be seeded by a set of known interactions performed by analysts. We plan to undertake such an elicitation study and expand on the details of this interaction space in future work.

Finally, analysts may be able to glean additional insight from their data by seeing the visualization from all three tools simultaneously, or by having a single tool that permits the analyst to switch between the three views. If the projection and clustering is relatively stabilized between views, or if smooth transitions and additional interactions are provided to enable analysts to efficiently locate observations and clusters across projections, the additional perspective acquired from each of the three visualizations can support the sensemaking process of the analyst.

8.5 Conclusion

In this work, we describe and evaluate tools that combine dimension reduction and clustering algorithms together into a single analytical system. However, the three tools that we have developed each treat the order of the algorithms differently: Gemini processes both algorithms independently in the high-dimensional space, Castor uses the output of the dimension reduction algorithm as input to the clustering computation, and Pollux uses the computed cluster memberships to influence the dimension reduction layout. By interacting with these tools, an analyst can explore a dataset and incrementally formalize hypotheses and conclusions regarding what they see in the data. Each of these three tools comes with

their own strengths and weaknesses, presents data in different layouts depending on the dataset used, and balances the tradeoffs between stress in the projection and intra-cluster quality in different ways.

Chapter 9

Cognitive Dimension Reduction and Clustering

The three tools developed in Chapters 6-8 represent three individual points in an immense design space of algorithms, tasks, visualizations, and interactions that can be applied to dimension reduction and clustering for sensemaking. Before performing a comparative study on these three tools, a broader and more interesting question to address is “How do analysts think about grouping (clustering) and spatial (dimension reduction) operations?” This overarching question incorporates a number of points for investigation, including understanding how analysts begin to explore a dataset, what types of cluster structures are created and what types of cluster operations are performed, what decisions analysts make when exploring individual observations, and how quickly the analysts explore the data.

In “The Semantics of Clustering,” Endert et al. perform an experiment to understand the clustering behavior of analysts performing a sensemaking task using LightSPIRE [115]. However, this study was based on a document collection rather than a numerical dataset, and participants were not strictly limited to spatial and grouping interactions with the data. This chapter contributes the design and results of such a modified study, in which a group of participants are asked to organize the observations contained within an unfamiliar quantitative dataset. Addressing this question will enable us to refine the tools we have created in order to better support analyst processes.

9.1 Background

Sensemaking refers to a cognitive process for acquiring, representing, and organizing information in order to address a task, solve a problem, or make a decision [182, 263]. A number of models with varying levels of information granularity have been proposed for approaching and solving sensemaking problems [247, 248, 263]. These models represent strategies for addressing a variety of sensemaking problems. For example, Pirolli and Card's Sensemaking Process [248] (see Figure 9.1) is designed for sensemaking problems faced by intelligence analysts. Despite the specific challenge addressed by each of these models, they all highlight the need to organize the data. For example, an intelligence analyst may work to understand the actors and motivations by grouping documents by location, by person, or by subplot.

A fundamental behavior in sensemaking is the act of grouping similar observations in order to understand their properties. This organizational strategy is true both in paper-based sensemaking tasks [101, 324] and in tasks performed on electronic displays [15, 115]. Clusters therefore have a natural connection to sensemaking. Clusters can also help to reduce clutter in a workspace, compressing similar observations into a group that requires less physical or screen space [215]. Simplifying the workspace leads to further cognitive benefits, as humans struggle to think about more than a small number of observations or dimensions at one time [274]. Thus, using groups of items to perform analysis tasks can lead to improved memory and recall by providing a simplified method of understanding the data [79].

Previous research has shown that humans use a variety of organizational principles to cluster information [96], even when addressing the same task [15]. In order to identify clusters computationally, hundreds of clustering algorithms have been implemented, each with strengths and weaknesses. As a result, there is no universally optimal clustering algorithm. Instead, the best clustering algorithm to solve a problem is often determined experimentally [118].

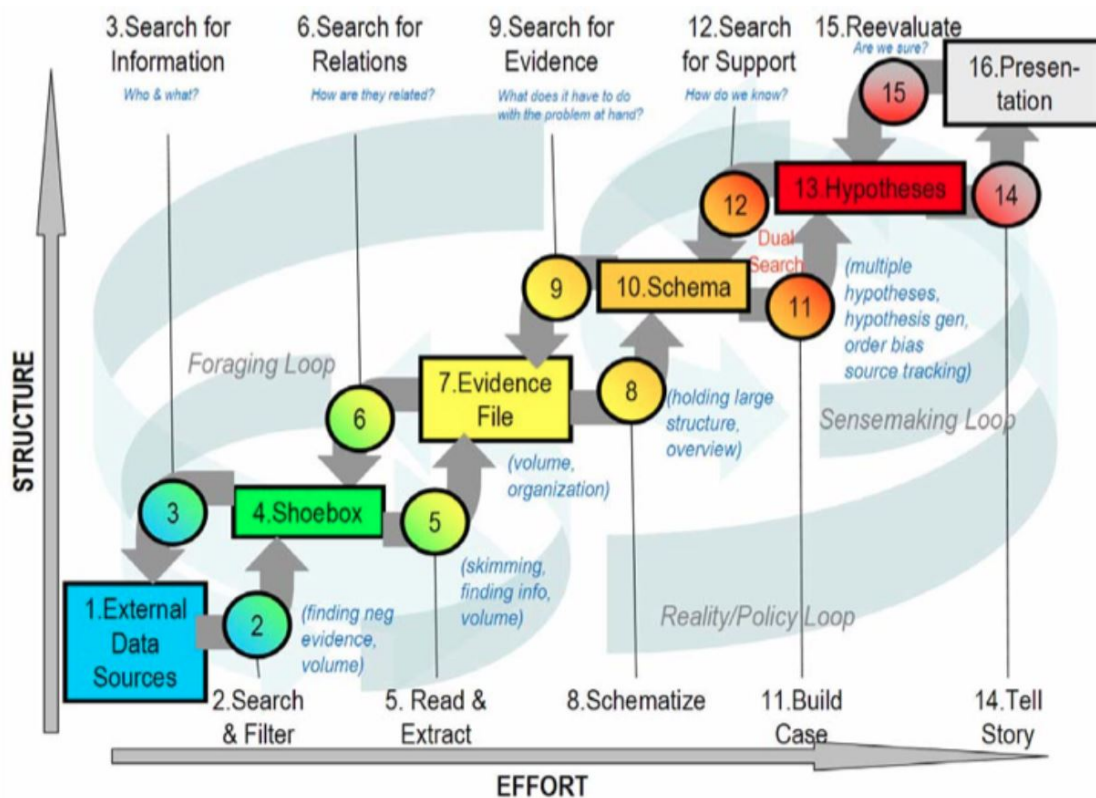


Figure 9.1: In the Sensemaking Process [248], intelligence analysts transform raw information into reportable results through organizational stages that filter, extract, and structure data. Included under Fair Use, 2019.

Distributed and embodied cognition share complementary roles in human sensemaking. Distributed cognition refers to the idea that external spaces can be used to extend and support cognitive reasoning. Analysts can thus use objects or symbols as a means of externally encoding relationships [255]. “Space to Think” demonstrated that space plays a meaningful role in sensemaking, providing a large high-resolution display grid to permit analysts to organize hypotheses and evidence spatially [15]. Spatial memory has also been leveraged by Robertson et al. for document arrangement in their Data Mountain system [254], and the role of spatial memory has been extended into 3D interfaces for information retrieval tasks by Cockburn and McKenzie [69]. Embodied cognition [326] focuses on the integration of the physical body and the environment with internal resources, reflecting how the body

Table 9.1: The dataset used in the cognitive study described in this chapter.

Animal	Furry	Big	Swims	Fierce	Solitary
Bat	39	1	0	33	20
Blue Whale	0	87	72	8	26
Bobcat	65	20	0	72	59
Cow	32	68	0	3	5
Deer	52	40	0	0	7
Giraffe	17	78	0	1	10
Grizzly Bear	82	87	3	62	59
Hippopotamus	0	79	40	19	25
Killer Whale	1	91	91	38	16
Leopard	43	55	5	70	34
Moose	37	88	3	9	32
Otter	47	11	85	6	31
Polar Bear	82	85	39	70	49
Seal	20	31	82	14	6
Squirrel	79	3	4	8	23
Walrus	19	76	77	20	11
Wolf	67	40	0	76	41

influences cognition. Embodied cognition allows analysts to offload cognition and create understanding within their workspace, allowing physical navigation to provide more meaning to locations [14]. “Space to Think” has been demonstrated to extend to more complex spaces which contain multiple displays and devices [66, 67, 141].

9.2 Experimental Design

The overarching question that motivates this research component focuses on the organizational processes of analysts when performing exploratory data analysis. We wish to understand the cognitive processes that underlie the approach that analysts take when trying to find insight from an unfamiliar dataset. In this study in particular, when analysts are only

afforded grouping and spatialization actions, how will they organize a collection of observations? In particular we wish to understand how analysts begin to explore a dataset, what types of cluster structures are created and what types of cluster operations are performed, what decisions analysts make when exploring individual observations, and how quickly they can explore the data. This study was designed to investigate these components of the exploratory data analysis process.

9.2.1 Participants

We recruited 11 participants from engineering and technology disciplines in an academic setting. Using a between-subjects design, we divided the participants into two approximately equal groups, each performing their organizational tasks on a separate set of observations. The group that received the labeled dataset (participants referred to as L1–L5) received 17 index cards with an animal name and five dimensions that describe the animal (see Figure 9.2 left). The second group received the same data in abstract form (see Fig 9.2 right; participants referred to as A1–A6), with all animal-related contextual information removed from the cards. The attributes are integers scaled to a 0–100 range. The complete dataset is provided in Table 9.1. Participants were asked to ignore the large numbers on the cards; they were added for better video recording as seen in other studies [153].

9.2.2 Dataset

The animals dataset provided to the participants is a reduced version of that created by Lampert et al [195], selected because of its general knowledge applicability to all potential participants. From the initial dataset, we rounded all decimal values to the nearest integer, and then reduced the number of animals and the number of dimensions. We selected these

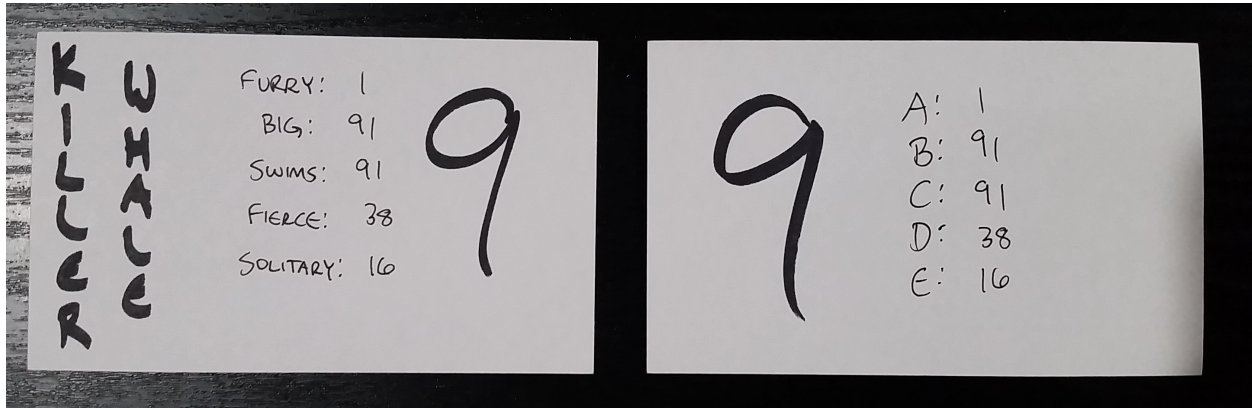


Figure 9.2: A photo of the Killer Whale card in both the **(left)** labeled and **(right)** abstract datasets.

17 animals with the foreknowledge that they could be naturally divided into three groups of five animals plus two outliers, but that alternate classifications and cluster assignments were possible. Five dimensions were selected to make the task challenging but not overwhelmingly difficult, with one dimension selected to describe each of the three groups and two additional noise dimensions. Thus, a proposed division of this dataset (displayed in Figure 9.3 with additional hierarchical clustering) could be:

- **Predators** (described by Fierce): Bobcat, Grizzly Bear, Leopard, Polar Bear, Wolf
- **Aquatic** (described by Swims): Blue Whale, Killer Whale, Otter, Seal, Walrus
- **Large Herbivores** (described by Big): Cow, Deer, Giraffe, Hippopotamus, Moose
- **Outliers:** Bat, Squirrel

However, alternative natural groupings are possible. For example, the Polar Bear and Hippopotamus have Swims attributes that could place them in the Aquatic cluster, or the Killer Whale could be a Predator (and the Bat has a similar Fierce attribute). The animals could be divided into two groups rather than three: Aquatic and Non-Aquatic, or into Furry and Non-Furry.

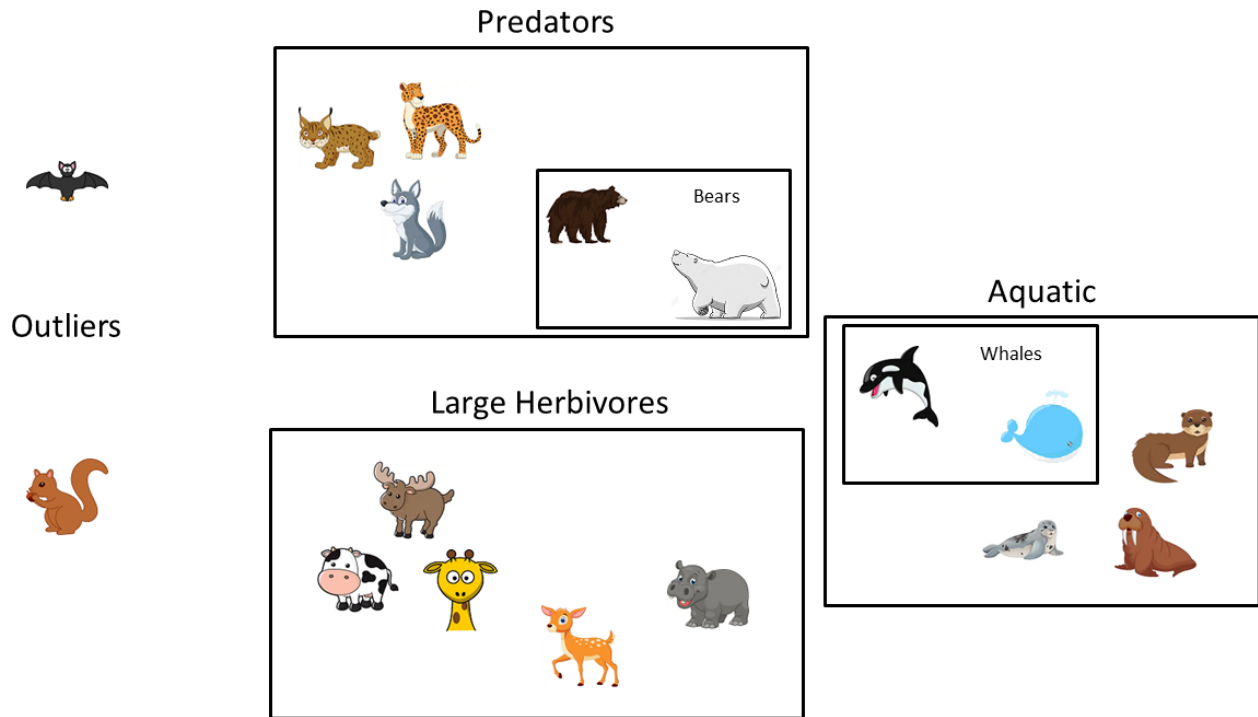


Figure 9.3: One of many potential organizational structures possible to generate from the study dataset.

9.2.3 Procedure

Noting the cognitive strain experienced by participants during a pilot study, we elected to limit the length of the study to one hour in order to minimize tiredness and frustration effects. As a result, several participants only completed the first organizational task, though many did complete both tasks. Participants began the study by responding to four questions in a Google Forms survey, describing their self-stated familiarity with dimension reduction and clustering algorithms and answering two questions about exploratory data analysis. All participants reported some degree of familiarity with clustering algorithms, but three of the participants reported no exposure to dimension reduction algorithms.

Following this, they were provided with a short description of the tasks they must complete and the goals of the study, after which they saw the dataset for the first time. In the first

task (referred to as Organization Task), participants were asked to organize the observations in any way that they wished, though they were limited to grouping and spatialization operations (i.e., “place two observations in the same group” and “place two observations some distance apart”). Participants were instructed to think aloud in order to better capture their organizational process [231, 336].

After completing this organization, the second task (referred to as Update Task) asked them to update their organization given new information that two of the dimensions were more important than the other three. These two dimensions were intentionally chosen as the dimensions that the participant used least in the Organization Task, in order to require a significant organizational update. Additionally, the participants who received the abstract dataset were informed of the animal mapping and asked to comment on their organizational structure given the addition of labels. Finally, participants completed a second survey of open-ended questions relating to their thoughts regarding their personal analysis process.

9.3 Organization Task Results

This section discusses results seen from the first 11 participants in the study. 5 of the participants received the labeled dataset, while the remaining 6 received the abstract dataset.

9.3.1 Beginning the Analysis

There were two primary methods by which participants approached the Organization Task. The most common strategy was to begin by laying out the full dataset on the table, often in a grid pattern, in order to inspect the full dataset simultaneously. Slightly less often, participants would keep the index cards in a stack, inspecting each sequentially and deter-

mining its optimal location in the partially-organized space. Several participants began by inspecting the top few cards in the stack before turning to one of the two primary patterns. A single participant who received the labeled dataset looked through the full stack of cards to survey only the names of the animals, organizing the cards in the stack before following the sequential pattern.

Interestingly, there was a divide between the common approaches in the two groups. Five of the six participants who received the abstract dataset followed the grid layout pattern, while only one of the five participants did so with the labeled dataset. Conversely, three of the five participants who received the labeled dataset followed the one-by-one pattern, while only one of the abstract dataset participants did so.

Only one of the participants in the abstract data group primarily used spatialization actions to begin exploration the data. This participant created a radial layout in which observations were drawn towards five points that corresponded to full values of each of the dimensions (Figure 9.4). The remainder of the abstract dataset participants performed mostly grouping operations to explore the dataset, as did three of the labeled dataset participants. The remaining two labeled participants performed a blend of grouping and spatialization operations, to the degree that neither category could be clearly considered a majority.

9.3.2 Cluster Structures

Both groups of participants were approximately equally likely to create cluster hierarchies and cross-cutting clusters in their organizational structures (2/5 and 3/6). Many participants did create internal distance structures within their clusters, but only a small subset clearly delineated clusters within clusters or clusters overlapping clusters (for example, Participant A3 in Figure 9.5). In many cases, these internal clusters were formed by breaking up a larger

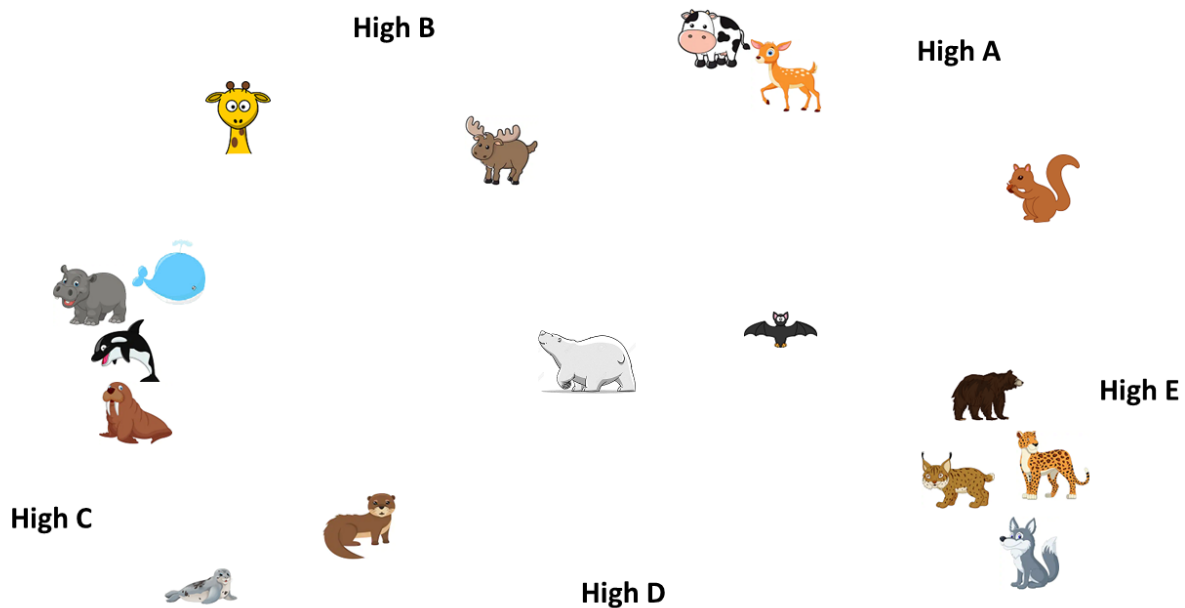


Figure 9.4: The radial layout created by Participant A2, in which each of the animals is drawn towards its highest attribute with additional effects by the other large attribute values.

supercluster, though occasionally two subclusters were joined together to form children of a larger parent cluster.

Both groups of participants were also equally likely to create organizational structures in which the axes mapped to dimensions in the data. This is contrary to the common behavior of dimension reduction algorithms in which the axes have no meaning. Often, such constructions resulted from participants' behavior in focusing on a single dimension at a time and organizing the observations on a spectrum along one or more dimensions (see Figure 9.6).

Both groups of participants were also equally likely to identify outliers in the dataset that they were hesitant to group in any cluster. Near the beginning of their analysis, they referred to single-observation groups as clusters (or the seeds of a cluster), but as they continued to structure observations in the space, they were more likely to refer to these observations as not fitting well with the others. The participants who created cross-cutting clusters often

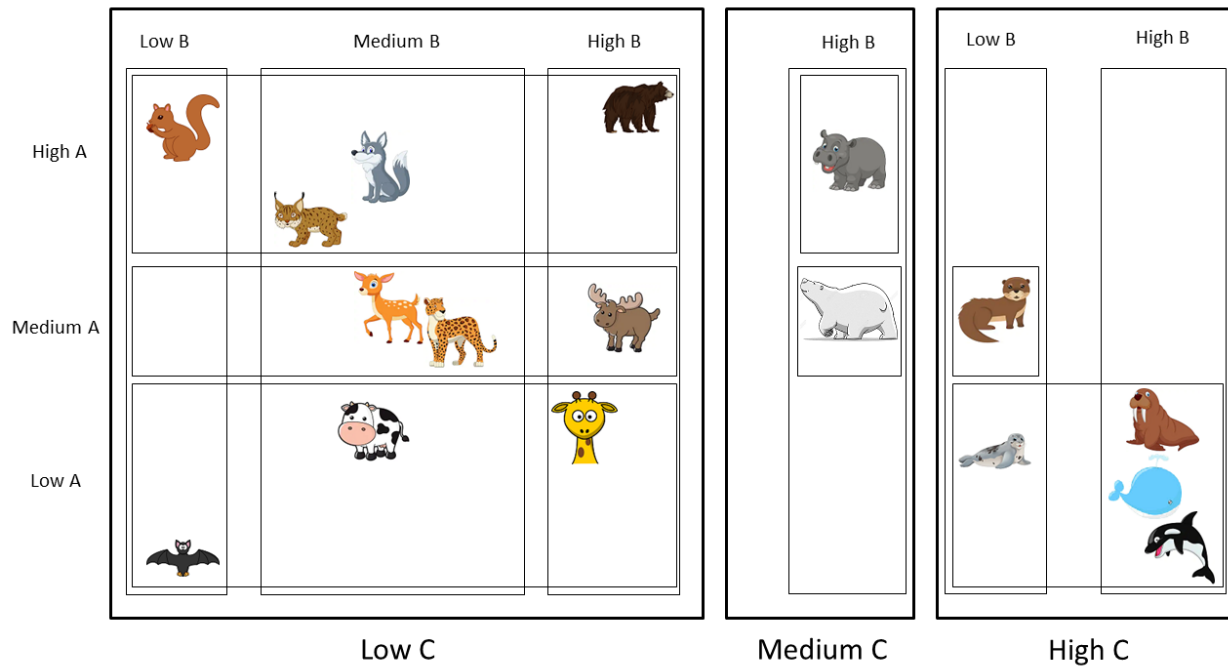


Figure 9.5: The complex cluster cross-cutting created by Participant A3. This participant judged clusters of abstract data by considering low, medium, and high values of dimensions A, B, and C (corresponding to Furry, Big, and Swims). The axes of the organization were important to the cluster determinations.

had subgroups with just a single observation in their organizational structure, but they were clear to identify those as equally belonging to two or more of the broader groups (again see Figure 9.5).

The sizes of clusters that the participants created was also quite broad. All but one of the participants had at least one cluster that only contained a single observation, and most of the clusters were small in cardinality. However, some participants did occasionally create large clusters encompassing 1/3 to 1/2 of the overall dataset.

Both groups of participants were also likely to create internal spatial meaning within clusters. Often, the spatializations within the clusters were designed to show differentiation within observations in the cluster (e.g., a size trend across the cluster), though occasionally the goal

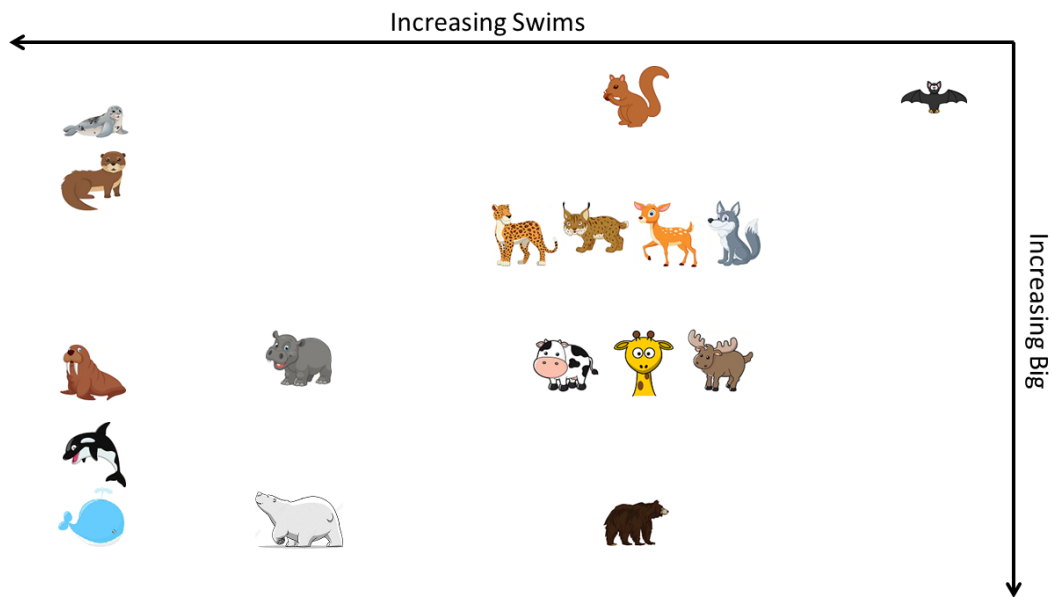


Figure 9.6: Participant L5 created a structure in which the axes were clearly an important feature. Groups were refined based on the remaining three dimensions, but the majority of the structure was governed by the Swims and Big dimensions with which she began her analysis. Attribute bins from the analysis of the Swims dimension are clearly still visible on the x-axis.

of the participants was to show relationships between members of the cluster and other parts of the space (e.g., an observation within the cluster that is quite similar to those in other clusters). As a consequence of the second, participants in both groups were equally likely to create global spatial structures that spanned the entire structure or governed large portions of their organization.

9.3.3 Cluster Operations

We recorded instances in which participants performed four different types of cluster operations: create, remove, join, and split. These are differentiated in Figure 9.7.

The primary method by which participants in both groups approached the organizational

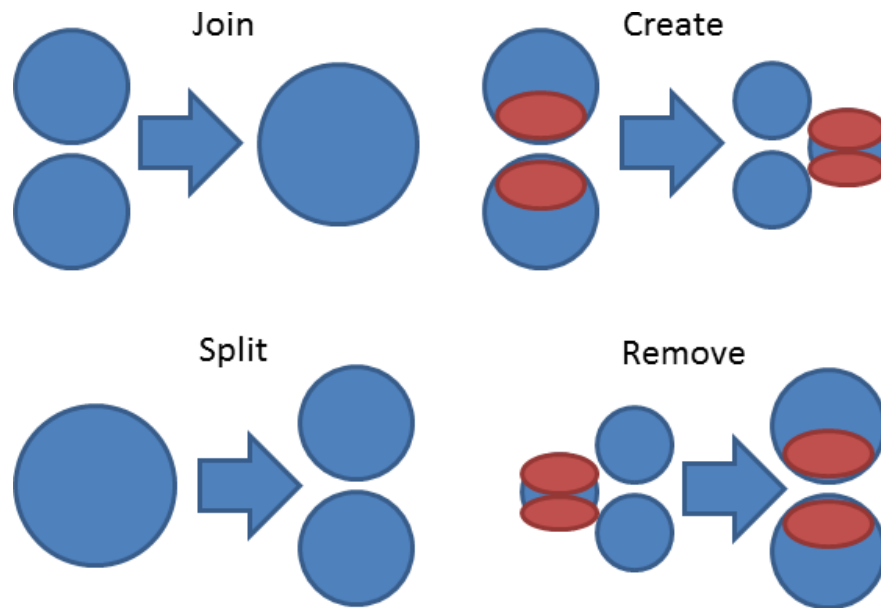


Figure 9.7: Cluster operations in the participant organizational strategies. Cluster join operations destroy two original clusters to create a new cluster, cluster split operations destroy one original cluster to create two new clusters, cluster create operations take a portion of one or more clusters to create a new cluster (while leaving the originals), and cluster remove operations destroy one original cluster and distribute its members to one or more existing clusters.

task was to start with large groups and then subdivide (the exception being A2 who created the balanced radial layout). As a result, splitting an existing cluster into smaller clusters was the most common cluster operation performed. A majority of participants also joined clusters together at some point in their analysis, usually when considering a dimension for the first time and noticing new similarities among the observations.

Only one of the participants in the abstract group (again A2) performed operations to create an entirely new cluster from observations previously in several other clusters. None of the participants in the abstract group removed a cluster and allocated its members into several other clusters. In contrast, three of the five participants in the labeled group created a cluster, and two of the five removed a cluster.

9.3.4 Decision Making

Again with the exception of A2, all participants in both groups spent the majority of their analysis considering only a single dimension at a time, confirming the observation seen in previous studies that analysts struggle to think high-dimensionally [14, 274]. Additionally, participants frequently processed attributes by either making binary decisions (e.g., divide observations by greater than or less than 50), or alternatively by creating a small number of bins to discretely group observations by a single dimension. The participants commented that both the binary decisions and the binning operations were intentionally made so that they could focus their attention on subsets of the observations rather than the entire collection.

Two of the participants in the abstract group created features from combinations of provided features while exploring the data, in both cases to reduce the amount of information that they were cognitively trying to process. One of these participants computed the median of all five dimensions in order to perform an initial grouping, while the other computed the difference of the last two variables she had not yet considered. Two of the participants from the labeled group also created features, but these came from domain knowledge of the labeled animals instead. One of the participants introduced both flying and speed features into her layout, while the other created groups that incorporated the likelihood of finding these animals in a zoo or in Canada. This last participant, L1, also reported that she focused primarily on the animal labels, and did not consider the attribute values until making final refinements to the space.

9.3.5 Timing

The time required by participants to complete the Organization Task to their satisfaction varied broadly across both groups. Some participants in both groups were satisfied with their

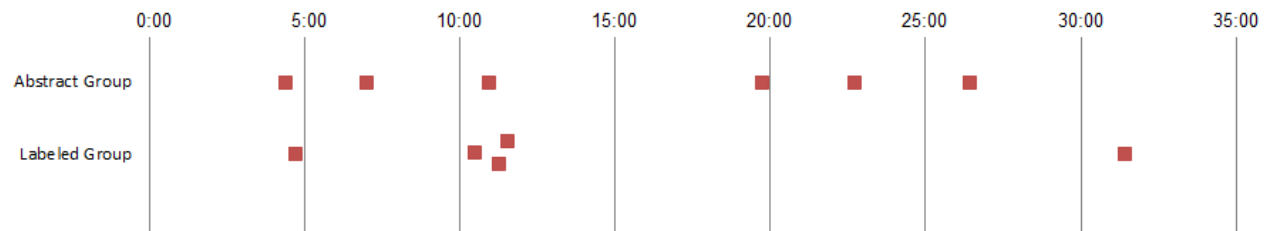


Figure 9.8: Time distribution for completing the Organization Task.

organization in less than five minutes, while others required approximately half an hour. The timing distribution for the groups is provided in Figure 9.8. Because of the limited number of participants evaluated thus far, it is difficult to conclude any trends from the data.

9.4 Update Task Results

Eight of the eleven participants had sufficient time remaining to address the Update Task. In general, there was no discernible difference between the behavior of the labeled and abstract groups when performing this task. This can be attributed to both the task being simpler with the smaller number of important dimensions, as well as the smaller sample size at this point in the study.

9.4.1 Beginning the Analysis

Much like with the Organization Task, there were two primary methods by which participants began their reanalysis. The most common strategy was to completely erase the existing structure, pulling all of the index cards into a single pile for future analysis. Next, they began to deliberately create a structure resembling a scatter plot, using the two important dimensions as axes and processing one observation at a time as they positioned data spatially. An example of a scatter plot created by participant A5 is shown in Figure 9.9.

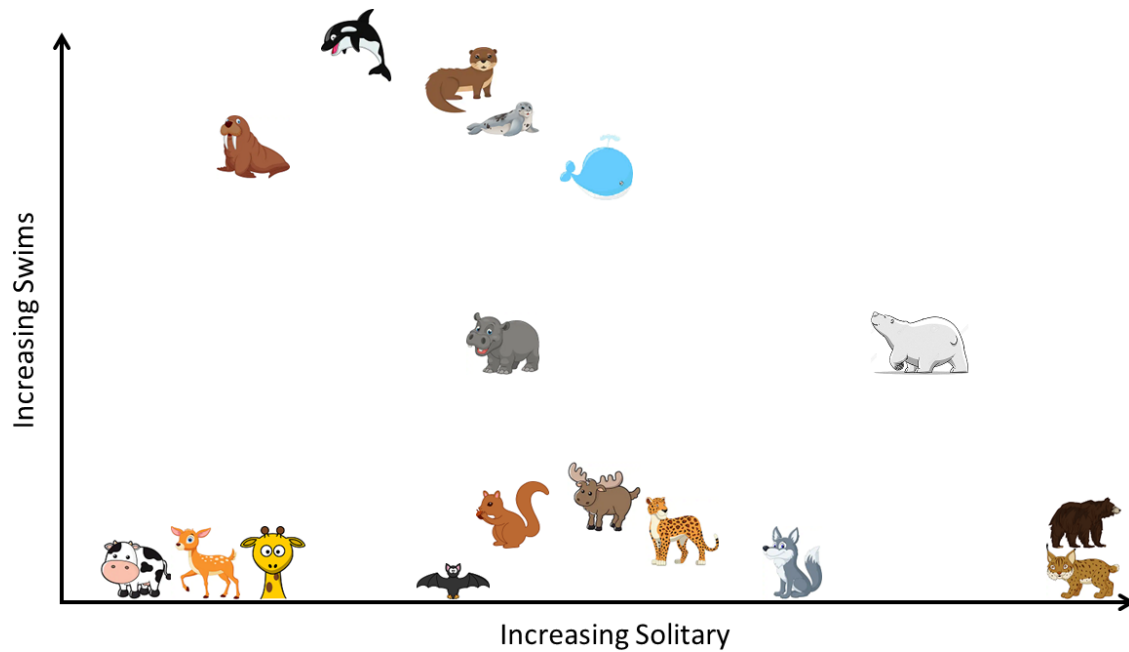


Figure 9.9: The scatter plot created by Participant A5 between the C and E (Swims and Solitary) dimensions.

The other common strategy was to leave the observations in their existing groups from the final state of the Organization Task, beginning to make incremental refinements to the structure to reflect the new importance information. Often, this strategy incorporated binary decision making, modifying the existing groups by shifting observations into high/low values of the important attributes.

The exception to both of these strategies was participant L2, who inadvertently happened to receive two dimensions that were somewhat correlated (Furry and Swims). As a result, this participant created more of a linear spectrum with high Furry and low Swims at one end, low Furry and high Swims at the other, and mid-range Furry and high Swims in the center. There was some two-dimensional structure to potentially interpret this as a scatter plot-like structure, but this was a result of identifying and creating groups in the observations while creating the spectrum rather than constructing a two-axis plot. This construction is shown in Figure 9.10.

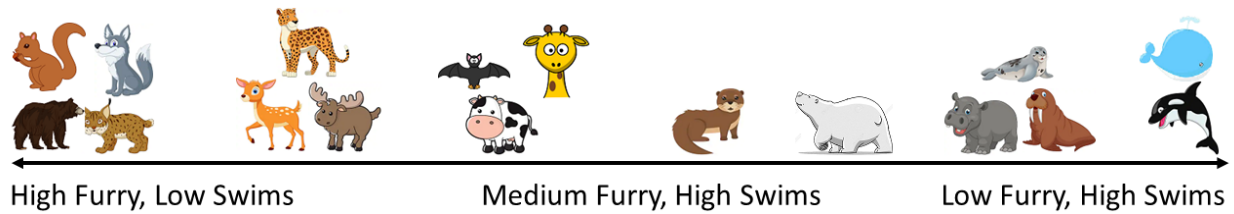


Figure 9.10: The spectrum of Furry and Swims created by Participant L2.

9.4.2 Cluster Structures and Operations

As participants addressed this task, they often performed more spatial actions than grouping actions. As a result, their final organizational structures had looser, poorly-defined groups. Participants occasionally would identify hierarchical structures within their space, but these were often limited (e.g., “there is a tight Whales cluster within the larger Aquatic cluster”).

As a result of the common scatter plot construction, axes were more likely to matter in the organizational structures created by the participants. Further, participants universally did not remove or split clusters as a result. Instead, they often created their initial scatter plots and then subsequently created and joined clusters.

9.4.3 Decision Making

Again as a result of the general scatter plot structure that participants created, there were many more global spatial structures created when performing this task than in the Organization Task. Only one of the participants (L3) created features for her analysis. Binary decision making and attribute binning were also a commonly-observed behavior, particularly early in the participants’ organizational strategy as they were processing a larger number of observations.

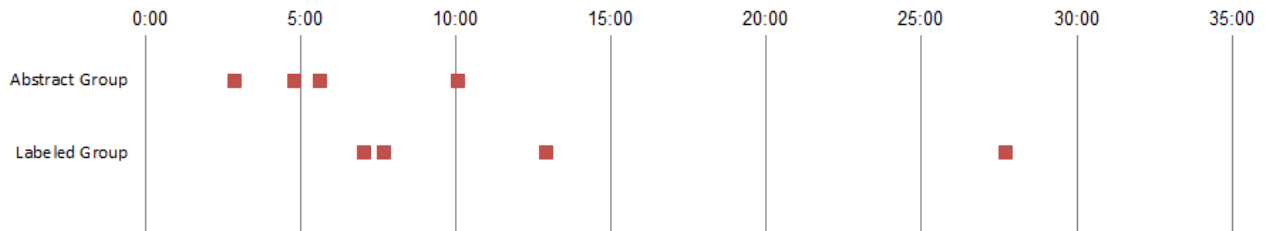


Figure 9.11: Time distribution for completing the Update Task.

9.4.4 Timing

As a result of only having eight participants taking part in this task, it is difficult to draw conclusions from the timing data thus far. However, there is currently a pattern in which the abstract group are generally performing the task more quickly than the labeled group (see Figure 9.11).

9.5 Discussion

It is difficult to begin to draw firm conclusions from the study given the limited number of participants to date. However, the results collected thus far do hints towards some useful clues for how humans approach exploring and organizing data, clues that can be used to guide the design of interactive visual analytics systems in the future.

9.5.1 Overarching Strategies

One of the most important results collected thus far is the tight coupling seen between spatialization and grouping actions performed by analysts. Rather than adopting a purely cluster-first or layout-first mentality, participants in this study switched between the two frequently. This was even true in cases where grouping or spatialization actions accounted

for a great majority of the overall interaction total. Further, we note that this complex relationship develops over time, where spatializations are used to drive clustering and clustering is used to drive spatializations. This results in complex organizational spaces that were produced by the participants in this study. We delve into these issues further in this section.

There were three main strategies that participants took when approaching the Organizational Task. The most common strategy seen is the **Divide and Conquer Strategy**. After spreading out all of the data, participants would select a single dimension and separate the observations into a small number of groups. They would then attempt to find meaning within these smaller groups, either by selecting another dimension to separate or by spatializing within the group. After structuring the individual groups, they would then turn their attention to the full space and attempt to organize the large groups, with occasional refinement within the groups.

An example of this strategy is provided by Participant L4 in Figure 9.12. In Panel A at 4:47, she has binned the animals by size, creating seven temporary groups that increase in size from left to right across groups and from bottom to top within groups. Panel B at 9:04 includes a dimension for Swims, forming a structure that approximates a scatter plot. At this point, she identifies three main groups: swimming animals, sort-of swimming animals, and non-swimming animals. In Panel C at 16:46, she has decided to separate the swimming animals cluster as she began to sort within the groups by the Furry dimension. The global Swims dimension still persists from bottom to top, but the global Size dimension is now discrete across the two columns of clusters. A local Size dimension was maintained vertically in the non-swimming animals group. The Furry dimension was vertical in the big swimming and sort-of swimming groups, but was horizontal in the non-swimming group (and was not clearly specified) in the small swimming group. In Panel D at 21:03, the two

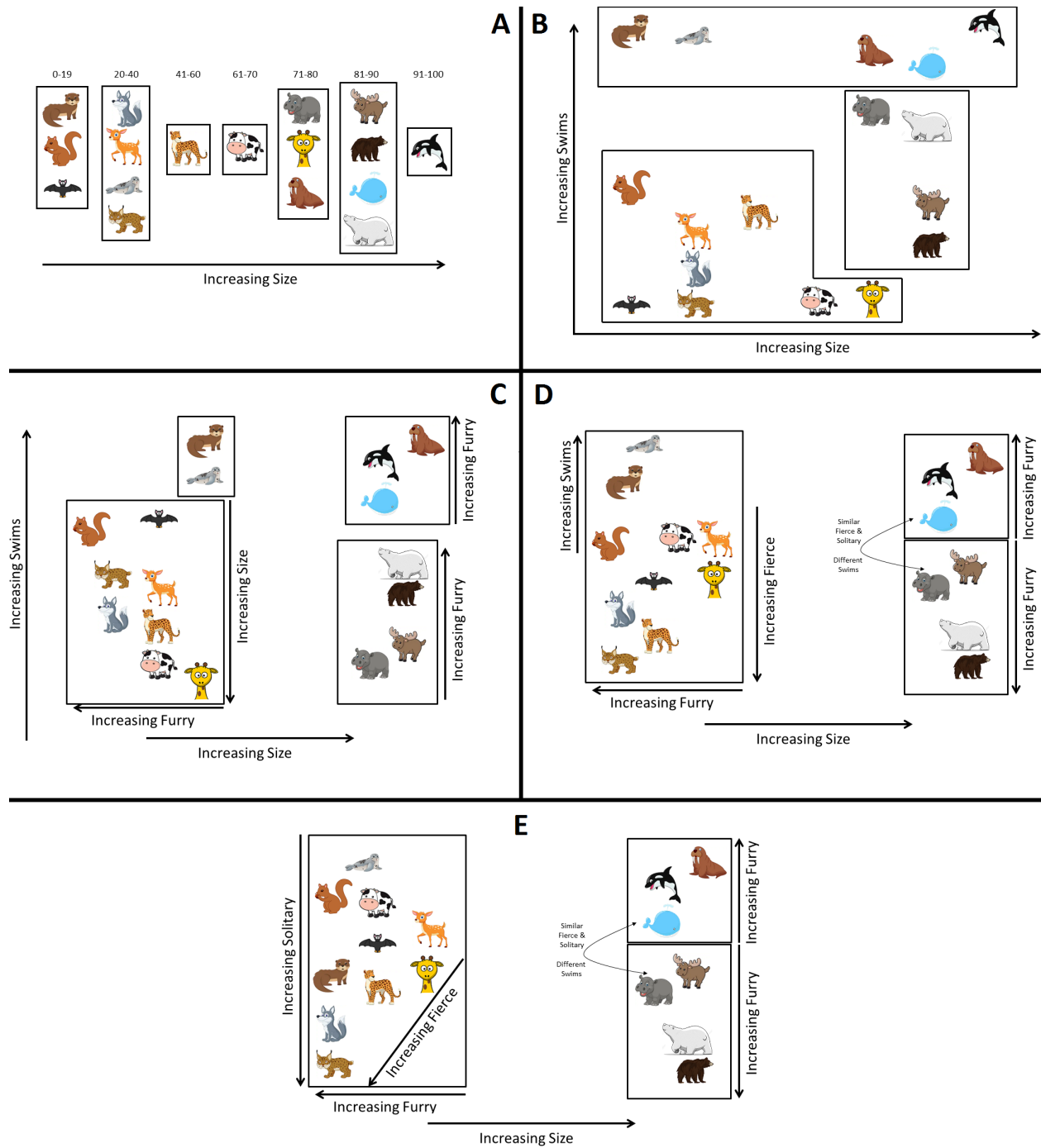


Figure 9.12: Five stages from the analysis produced by Participant L4.

groups of large animals were positioned closer together (though kept as separate clusters) and the sort-of swimming animals group was flipped vertically because the Blue Whale,

Moose, and Hippopotamus have similar Fierce and Solitary attributes. The vertical axis of the smaller animals cluster (joined into a single cluster) now has a Fierce dimension with the non-swimming animals, though the Swims dimension is still maintained at the top of the group. Finally in Panel E at 31:21, the Fierce axis was rotated within the small animals cluster, and the vertical axis has been replaced with a Solitary dimension.

The second most common strategy was the **Incremental Layout Strategy**. This strategy was almost exclusive to the labeled data group (A2 once again being the exception). Participants would consider each observation one at a time, adding them to a continually growing organizational space in the place which appeared most sensible. Each of these additions was often a grouping operation, but could also be a spatialization operation in some cases. Updates to the position of observations already positioned did occur, but were infrequent. As the participants continued to add data, the physical size of the space utilized increased. After all observations were added to the space, the participants began a more thorough refinement process.

A third strategy, not as common as the first two but still implemented by multiple participants, was the **Bottom-Up Strategy**. Participants would begin similarly to Divide and Conquer, laying out all of the observations to view simultaneously. However, their next step was to begin to build small groups of two or three similar observations, usually by only looking at a single dimension at first but then considering others. After many of the observations had been placed in small groups, spatial relationships were created between the groups, often leading to the formation of larger groups.

9.5.2 Post-Survey

All participants provided responses to the post-survey that followed this study. Each participant responded to questions about their interpretation of their strategy, the easiest and most difficult parts of the analysis, and their thoughts on the usefulness and meaningfulness of grouping and spatialization actions.

Participants reported approaching both tasks by primarily considering a single dimension at a time, confirming observations from the study. The difficulty that the participants experienced when attempting to think high-dimensionally suggests the need for computational support in similar organizational tasks. Each dimension was selected by searching for features in the dataset that seemed to be most useful or representative, either due to the overall distribution, outliers, or common values. When considering each dimension, participants sought out commonalities between the observations, building large groups or binning the observations in a spectrum and refining the bins. Participants who received the labeled set also used their own knowledge of the animals to create and organize groups.

When asked whether they thought they performed more grouping or spatialization interactions, participants gave a variety of responses. Among those who believed they performed more grouping operations, they noted that their overarching strategy was to isolate groups within the data in order to make future processing simpler. Those who believed that they performed more spatialization interactions generally reported that they were careful when refining the organization in later stages, leading to that majority. Some participants reported that they couldn't determine which set of actions was the majority.

Participants reported that the easiest part of the Organization Task was the beginning or the end of the analysis. Some reported that selecting a starting point, picking the initial groups, or performing the initial spatial structure was easiest. Others noted that making

final refinements within and between clusters at the end of the analysis was easiest. Fittingly, most participants reported the mid-stages of organization to be the most challenging, when they had to update existing groupings of data due to examining a new dimension and to keep existing spatial relationships when updating other relationships. Participant A1 did report that the amount of data seen after laying out all of the cards initially was overwhelming, but that did not stop them from arbitrarily selecting a starting dimension for analysis.

Participants were also approximately evenly split between considering the grouping or the spatialization actions to be more useful or more meaningful. Those who felt more positively about the groupings mentioned summarizing the big picture and making sense of the overall space visually, while those who felt more positively about the spatializations thought that these actions made them more careful in their analysis. Both groups also mentioned that their operation preference for this question impacted the other. Those who felt more positively about the grouping actions noted that it made spatialization actions easier to perform, while those who felt more positively about the spatialization actions noted that it made the task of creating meaningful groups easier.

A related observation while running the study is the role of terminology, particularly with the grouping operation. There were a number of times when participants were clearly separating the observations into piles, but they were somewhat hesitant to define this organization as a “cluster” or a “group.” Frequently, we found ourselves using a variety of terms when inquiring about the structures that the participants were creating (e.g., “Do you consider this a group, or is it a cluster, or an organizational construct, or a bin, or a collection, or ...?”). To the participants, the terms “cluster” and “group” seem to have a different, deeper semantic meaning than a simple “pile” of observations. In order to be classified as a “cluster” or “group,” participants often wanted to perform multiple iterations of analysis to ensure that more than just a single property defined the collection of observations.

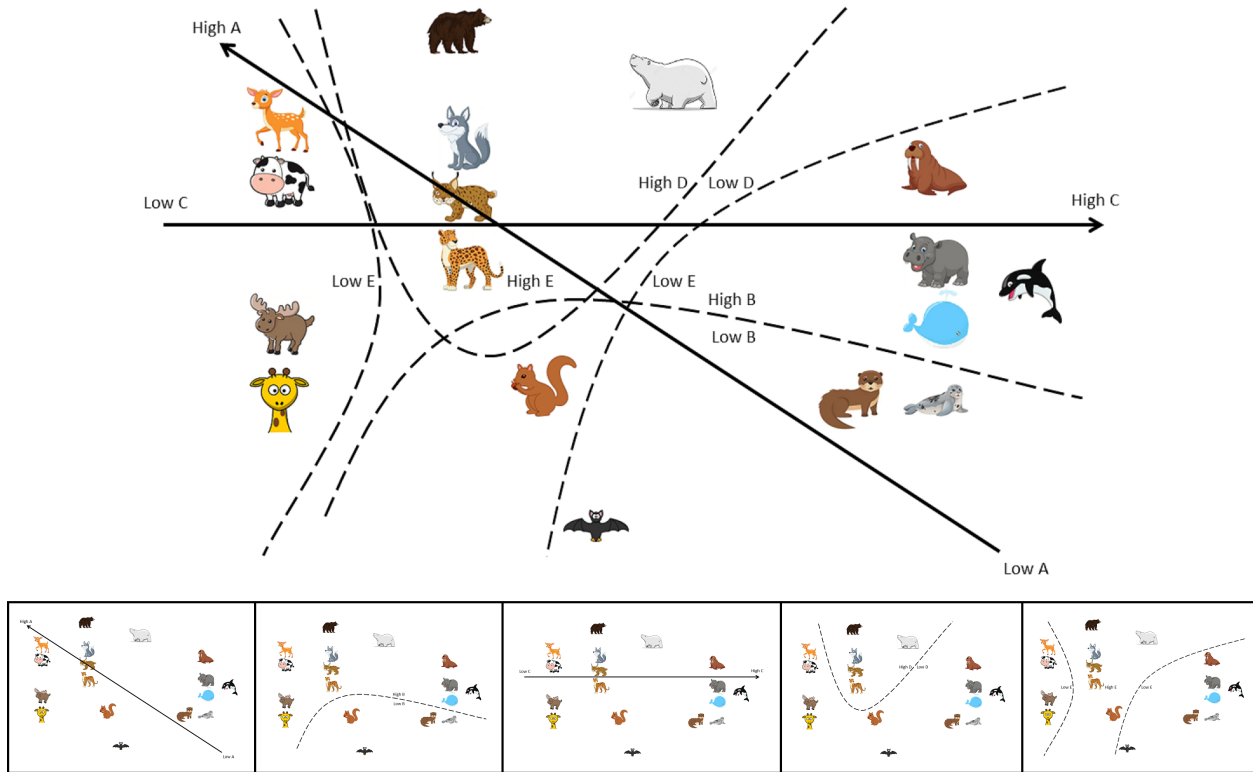


Figure 9.13: Participant A4 created a complex space, creating spectra for dimensions A and C (Furry and Swims) and distinct regions of high influence for dimensions B, D, and E (Big, Fierce, and Solitary).

9.5.3 Complex Spaces

After considering several dimensions, participants began to create complex spaces. This was already seen through the hierarchical, cross-cutting set of clusters created by Participant A3 (Figure 9.5). Another example is seen within the structure created by Participant A4 (Figure 9.13), in which the participant created spectra for two of the dimensions and regions for the three remaining dimensions. The two spectra were not orthogonal, though the C dimension was aligned with the x-axis. The three region dimensions likewise overlapped in some places but not others in the overall space.

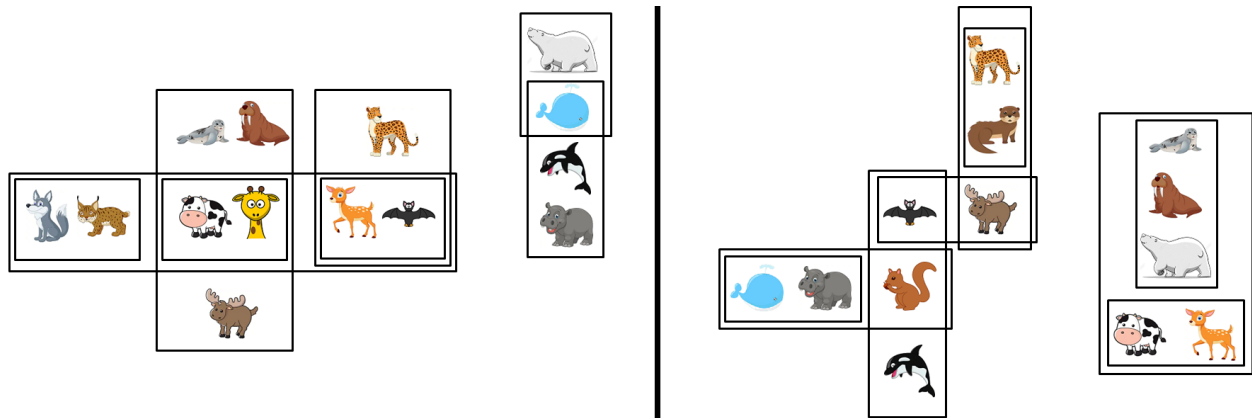


Figure 9.14: Participant A6 created complex hierarchical and cross-cutting cluster structures during her analysis.

This runs counter to the common method of creating dimensionally-reduced tools, in which the entire space is governed by a single weight vector (for example, [38, 98, 272, 317]). Creating clusters that contain independent weight vectors within, as well as a global space, presents one solution to this challenge. For the example presented by A4, the low B region could be defined as a cluster that still maintains the influence of low A and high C. This finding presents opportunities for the introduction of subspace clustering techniques (see Section 3.1.5) to support such complex spaces.

Further, these complex spaces are not limited to areas of attribute influence. Participants also often created complex structures of hierarchical, cross-cutting clusters. For example, Participant A6 created several complex sets of clusters at various points in her analysis, two of which are provided in Figure 9.14.

9.5.4 External Knowledge

Participants who received the labeled dataset often made use of their external knowledge about the animals that was not contained on the cards. This influenced their organizational spaces in three general ways. First, introducing external knowledge allowed the participants

to begin forming groups of animals prior to closely examining the data. For example, Participant L1 created a number of groups without even considering the data, including features corresponding to environment, diet, and probability of locating the animal in Canada. Likewise, Participant L5 introduced a flight group for the Bat into her organization, keeping it separate from all other animals as a result without considering the attribute values on the card.

Second, external knowledge was used to structure and spatialize within groups. For example, Participant L5 introduced a speed dimension into her organization, which permitted her to break up the land-dwelling (and non-flying) animals into groups based on how quickly they moved. The final result of this dimension introduction was a group that contained the Deer along with the Wolf, Leopard, and Bobcat. Occasionally, incorrect domain knowledge could also affect their organization, as seen when Participant L3 used external knowledge about the diet of animals to create transition groups that joined clusters. For example, the Hippopotamus was used as a transition animal between Aquatic and non-Aquatic animals not because of its mid-range Swims score but because of the fact that she felt this animal was likely to consume fish while spending time in water as well as plants when spending time on land.

Third, external knowledge about the animals was used as confirmation, checking to see if the groups the participants created were sensible after performing a dimension-specific organizational step. Participant L2, for example, frequently questioned his positioning of the Bat throughout his iterative organizational steps. Despite the attributes on the cards supporting his decisions, he continued to second-guess its position and occasionally to modify it based on his external knowledge. Participant L4 also reported scanning the names of the animals in the groups regularly, searching for refinements that could be made to the structure internal to clusters.

9.5.5 “The Big Reveal”

Participants who received the abstract dataset were informed of its animal dataset source after completing their organization. The animals cards were overlaid above the abstract cards, and participants were asked to comment on the structure and relationships that were now apparent after seeing the labeled data, effectively replicating the confirmation use from the previous subsection.

In general, participants reported being satisfied with the layout that they created in the abstract data. Many had an Aquatic group of animals due to the high-C score, and they quickly picked up on this relationship. Similarly, participants who prioritized the D dimension saw clusters of predators in their organization. The Bat often was the sticking point for participants, who were dissatisfied with its position in the space, similar to the reaction seen by participant L2.

When asked what they would change in their space given this new information, most participants increased the density of the clusters in their space, moving groups of similar animals closer together after understanding their relationship to each other. This was often seen in any Aquatic groups, Predator groups, and among the Whales. Interesting, this behavior was not necessarily true with the Bears; participants were more willing to keep these animals separate and to occasionally increase their separation despite the common genus, often reporting that this was counter to their initial reaction when noting the existing of two bears.

9.5.6 Computational Aid for Complex Visualizations

The study performed in this chapter intentionally used a small dataset to make the task achievable in a short time for human participants. However, only having five dimensions in the dataset may have influenced the outcome of the study. It is much easier for a par-

ticipant to create a space with five dimensions of influence than 50 or 500 dimensions, and so participants may have created spaces more complex than necessary to best represent the data. Further, the order in which the participants selected dimensions to include also affects the layout, as earlier dimensions set the overall layout that is refined by later dimensions. The fundamental cognitive limitation of participants evaluating a single dimension at a time plays an additional role here.

As a result, computational support could be introduced to better help participants to organize their data. For example, computationally identifying correlations between dimensions can enable participants to select one dimension and skip another. Identifying dimensions with substantial variance that could play a role in separating groups of observations can also be performed more efficiently by machines than humans (as could the median calculation performed by Participant A5). Further, participants could interactively probe their organizational space to uncover how subsets of dimensions interact with each other as they progressively make their space more complex. Each of these proposed techniques can improve the scalability of this study, thereby enabling the study results to be better mapped to large-scale datasets.

9.5.7 Lessons for Future Tool Development

Several observations noted in this study can be used to better design tools and visualizations that match user tasks and expectations. For example, nearly all of the participants created spaces in which the axes mattered (Section 9.3.2), aligning one or more dimensions to the surface that contained their space. However, many dimension reduction techniques consider distances between observations as more important than axes. Even dimension reduction techniques that do make use of axes (e.g., PCA) map synthetic dimensions to the axes

rather than information directly from the dataset. Either including axes as a consideration in the visual layout or better communicating the role of each dimension within the projection should benefit the understanding (and trust) of the analyst.

We also noted a tight coupling between the spatialization and grouping interactions, in which groups were used to refine the spatialization at some points, whereas the spatialization was used to define groups at other points (Sections 9.5.1 and 9.5.2). Providing an analyst with visual feedback to suggest potential ways of combining these interactions can better support user sensemaking. For example, an analyst could get a preview of the new projection and clustering that would result from performing a repositioning interaction before the interaction is processed by the same. Seeing this future projection could enable the analyst to glimpse an underlying cluster structure that was not apparent before. Similarly, a preview of the effect of a cluster reassignment interaction could enable the analyst to identify correlations between dimensions or to see underlying structure in the data that was not previously apparent.

Further, the complex spaces that were created by the study participants often included areas in which some dimensions had higher influence than other dimensions (Section 9.5.3). These localized areas of influence would be difficult, if not impossible, to capture using a global weight vector, as is often seen in existing interactive tools. Instead of this global technique, tools should be developed that allow for areas of local refinement, enabling substructures to be uncovered and evaluated by analysts.

9.5.8 Limitations and Future Work

In addition to the low number of participants tested thus far, the primary limitation of this work is that it relies on a sample of convenience. Although we found that participants were generally well versed in dimension reduction and clustering for data analysis, a more

Table 9.2: A summary of the main findings uncovered by this study.

Section	Finding
9.3.2, 9.5.1	A tight coupling was seen between grouping and spatialization actions, with participants frequently switching between these operations.
9.3.2, 9.4.2	The axes often had meaning in the organizational spaces created by participants, with one or more dimensions frequently front parallel or orthogonal to the front of the table.
9.3.3, 9.5.1	A common trend was to progress from big groups to small groups, splitting clusters rather than joining.
9.3.4, 9.5.2	Participants primarily explored one dimension at a time, expressing frustration with trying to consider all dimensions at once. This suggests the need for computational support.
9.3.4	To further reduce the complexity of the data, participants often binned the observations into smaller groups or separated the observations with a binary decision.
9.5.2	Participants often formed groups in order to make future spatializations easier, and also formed spatializations in order to make future groupings easier.
9.5.3	Computational tools need to support more complex spaces, rather than allowing a single weight vector to express the full space.
9.5.4	Participants in the labeled group often brought their external knowledge into their organizational structure, adding additional animal properties that were not provided.

educationally-diverse participant set would lead to more generalizable results. This can be corrected with the addition of future participants in this study. Additionally, this study has only been tested on a single dataset rather than experimenting with datasets of various sizes (cardinality of observations and dimensions), types (documents), and levels of complexity (floating-point observations, conflicting dimensions, and confounding variables). Future studies with other datasets can continue to explore this space.

9.6 Conclusion

In this work, we experiment with a labeled and abstract set of data to examine how analysts approach and organize an unfamiliar dataset. We wish to understand the cognitive processes

that underlie the approach that analysts take when trying to find insight in data. We found that participants used groups to create spatial structures as well as spatial structures to form groups. Participants created hierarchies and cross-cutting clusters in their organizational structures, and frequently approached the task by creating large clusters and subdividing them to refine additional structure. The complex spaces created by participants hint towards structures that should be supported in interactive applications. We summarize a list of main findings in Table 9.2.

Chapter 10

Human in the Loop Research Agenda

Interactivity in visualizations and analytical tools provides substantial benefits to the sense-making processes of analysts, with domains for such tools ranging from investigative analysis of documents [285] to exploration of scientific simulation results [179]. Recent investigation in the visual analytics domain has begun to explore the benefits of tools and algorithms that learn the intent of an analyst from their explorations, thereby enabling a system to adapt the layout and contents of a visualization to reflect the user’s mental model [38, 157, 199].

A number of visual analytics tools provide interactive data projections that update based on learned user behaviors [194, 214, 225, 239, 317]. For example, Andromeda [272, 273] and Dis-Function [42] support exploratory dimension reduction for high-dimensional quantitative data. Scientific visualization introduces opportunities for learning from interactions, such as analysis of data ensembles [234], interactive visual querying of nonlinear solution spaces [71], in-situ analysis of large scale data, and data foraging through extreme scale data [315].

This area of research at the intersection of visualization and machine learning is still novel, and much research remains to be performed in a number of areas. In particular, research into semantic interaction has begun to investigate complex ways in which machine learning can be used to support visualization. This stands in contrast to Explainable AI, which considers methods by which visualization can be used to support machine learning (see Figure 10.1).

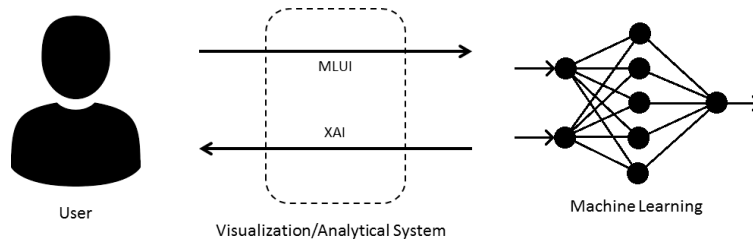


Figure 10.1: Explainable AI focuses on how visualization can be used to support machine learning (a human learns the machine state), while Machine Learning from User Interaction identifies ways by which machine learning can be used to support visualization (a machine learns the human state).

This chapter presents excerpts from a research agenda [322] to advance the Machine Learning from User Interaction (MLUI) field, generated through discussions at a workshop by a group of experts in this domain. We select challenges from throughout the feedback cycle found in Human-in-the-Loop systems that are applicable to the dimension reduction and clustering research contained in this dissertation. Learning from user interaction can assist human sensemaking in analysis [110, 274]. The excerpts in this section discuss challenges and opportunities where further research is needed, including needs for developer creativity to generate novel tools and user studies to better understand the effects of design decisions. This discussion is accompanied by descriptions of state-of-the-art tools and techniques that support research in this space, as well as ways in which the Gemini, Castor, and Pollux tools currently support or could be extended to support these research items.

10.1 Workshop and Research Agenda Background

The content that coalesced into this research agenda was generated by participants at the Machine Learning from User Interaction for Visualization and Analytics workshop at IEEE VIS 2018. The high-level goal of this workshop was to bring together researchers from across the visualization community to discuss how machine learning can be used to support

visualization tools and workflows. While many systems enable user interaction to explore data and parameter spaces in analytics, this workshop goes one step further to examine how systems can learn from user interactions to iteratively produce even more insightful results.

At this workshop, two sessions were designed to include a set of motivating papers followed by dedicated time for discussion in breakout groups. Approximately 50 attendees participated in these discussion sessions. Discussion groups were semi-supervised by the workshop organizers, and seeded with the discussion goal of identifying applications and open research questions related to the workshop topic. Each group included a note taker, who summarized the group discussion in real-time using a shared Google document.

Following the workshop, we collected these distributed notes into a single document in bullet point form. Within this overarching document, we began an affinity diagramming process to uncover common themes, research topics, and open questions. This affinity diagramming process was iterative, and we continued to refine topics and discussions in weekly meetings over a period of several months. The topics that resulted from these discussions (found in Sections 10.2–10.7) were then placed in a sequence that represents computational and user-driven components in an exemplary human-in-the-loop analytical system (Figure 10.2).

This research agenda captures five interconnected phases: an analyst interacting with a system (Interactive VIS of ML Data), the system logging those interactions (Capture Logs), the system learning from those interactions (Personalization), the system communicating its learned response to the analyst (Explainable AI), and the analyst interpreting that learned response (User-in-the-Loop Evaluation). Each of these phases can be roughly categorized by whether the analyst performs the given phase, the visualization/analytical system, or machine learning within the system. This division is denoted by the relative position of each of the phases in the central portion of this figure to the different phases. Each phase is explained in more detail in Sections 10.2-10.6.

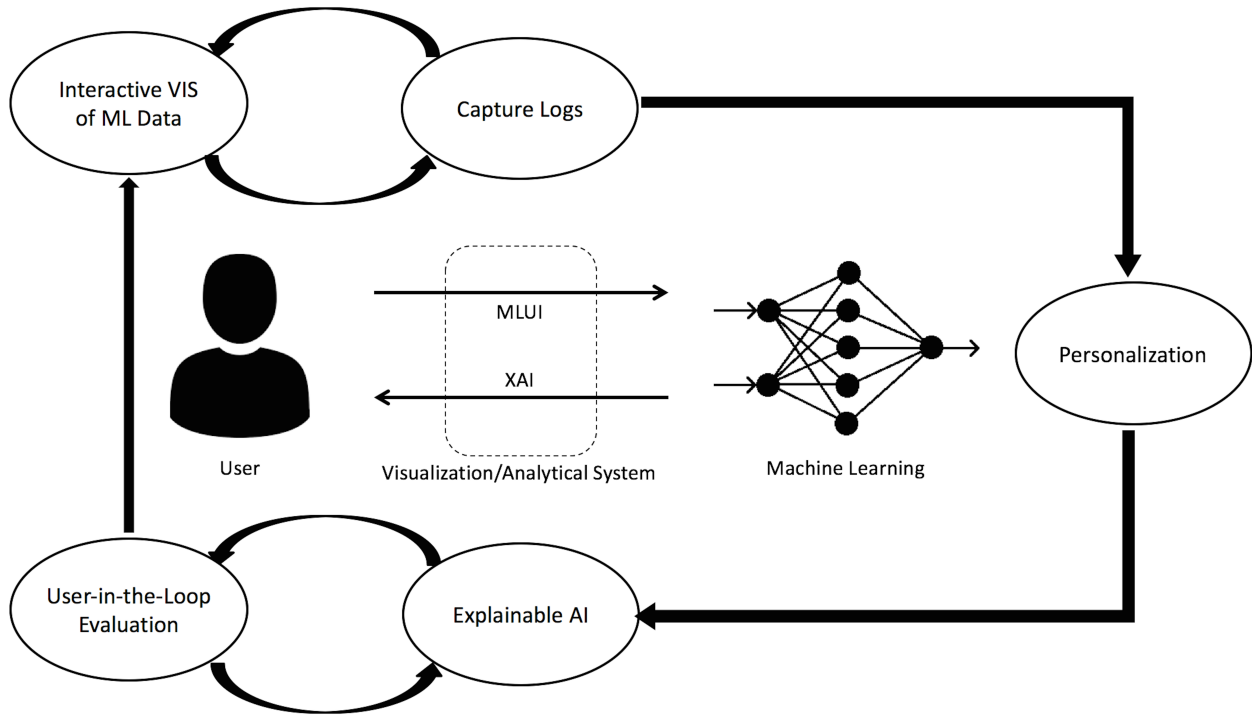


Figure 10.2: This research agenda captures five interconnected phases in an exemplary human-in-the-loop analytical system framework.

10.2 Interactive Visualization of Machine Learning Data

We begin our human-in-the-loop process discussion at the point where the analyst is about to interact with the system (the node labeled “Interactive VIS of ML Data” in Figure 10.2). The system has constructed a visualization of a dataset for the analyst to explore, and the analyst has evaluated the current visualization and determined how well it supports their analysis goals. This visualization could represent either the initial state of the system, or it could be at some later stage within the analysis process. In this section, we discuss how the analyst understands the interaction techniques available to them and determines which best supports their goal. Our discussion focuses on three issues: how should systems guide user interaction, what descriptions are available to allow the analyst to interpret these interactions, and where does the analyst even start? This user-centric understanding is tightly coupled with the

system capturing interactions to continue learning about the analyst, as discussed in more detail in the next section. These two phases form an iterative process in which the analyst may perform multiple logged interactions in order to reach a specific analytical goal.

10.2.1 Guiding Analysts Towards Interactions

To help the analyst understand what they can or perhaps should interact with, machine learning can be used to help guide the analyst towards an interaction. Two major research questions related to system guidance are **How should systems provide this guidance?** and **How will analysts interpret the effect of a recommended interaction?**

One form of system guidance is making use of recommendation algorithms to highlight information of interest to an analyst, thereby assisting them in focusing their exploration on relevant information. When considering how to accomplish such recommendations, an important question is, **Which interactions and data should analysts be guided towards?** This question remains a significant research challenge in which the answer largely depends on the system's perceived importance of data. The idea of information scent and scented widgets [59, 60, 61, 246, 325] embeds navigational cues and interaction effects into potential analyst actions. Such techniques could guide analysts towards interesting or unique data, recommend commonly-used interactions, or highlight attribute values that have not yet been explored, thereby resulting in additional insight on the data. Further, the system itself could request information from the analyst directly via an active learning approach, guiding the analyst towards interactions that the system believes will best improve the model [275].

A related question is: **What visual metaphor should the system use to convey this guidance?** For one example, StarSPIRE [38] and Cosmos [99] recommend documents (i.e., data to explore) to the analyst by filtering documents to display based on the perceived

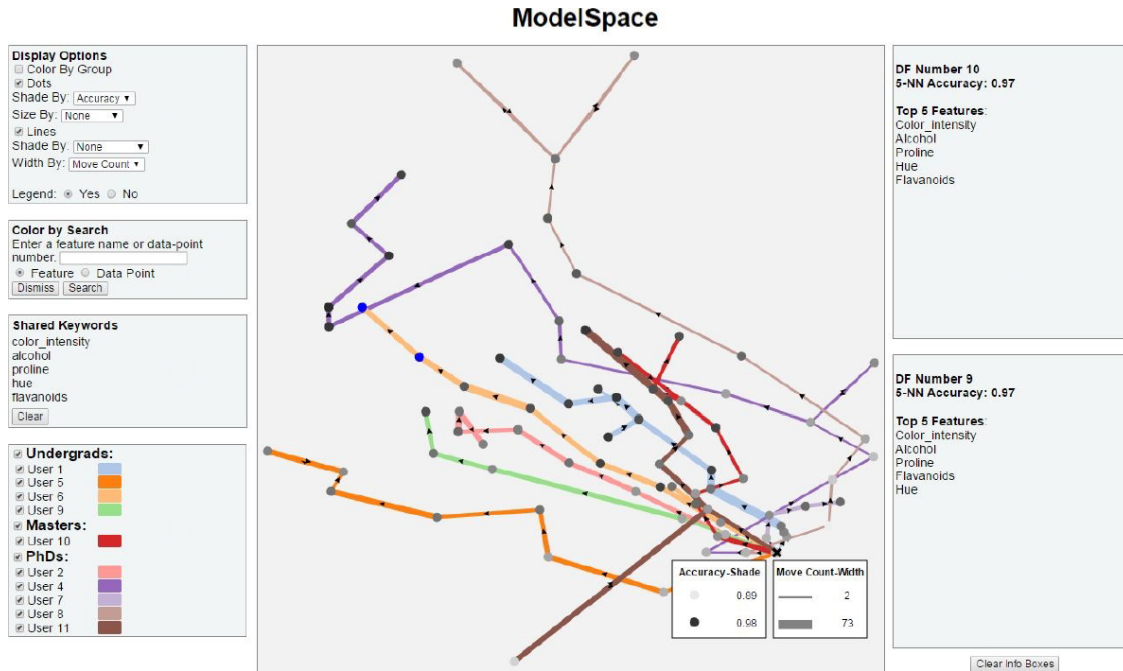


Figure 10.3: ModelSpace [44], displaying a layout of models and interaction paths. Included under Fair Use, 2019.

relevance of those documents to the analyst’s current analysis. This perceived relevance is mapped to the corresponding node sizes of the displayed documents, providing the analyst with more information regarding this recommendation that denotes which of those documents may be most relevant to them. Alternatively, ModelSpace [44] (see Figure 10.3) shows a projection of the system states that an analyst has already explored, providing a visual reminder of unexplored regions that should be evaluated for completeness of analysis.

Related to the previous questions, **How should the system guide user interactions in such a way that analysts will understand the potential effects of an interaction?** ML responses are complex and difficult to predict, which might cause fear of interaction. For example, analysts who are unfamiliar with complicated ML algorithms that are being used to create the visualization may believe an interaction will have a small impact. When the interaction results in a much larger change, the analyst may be confused since their mental

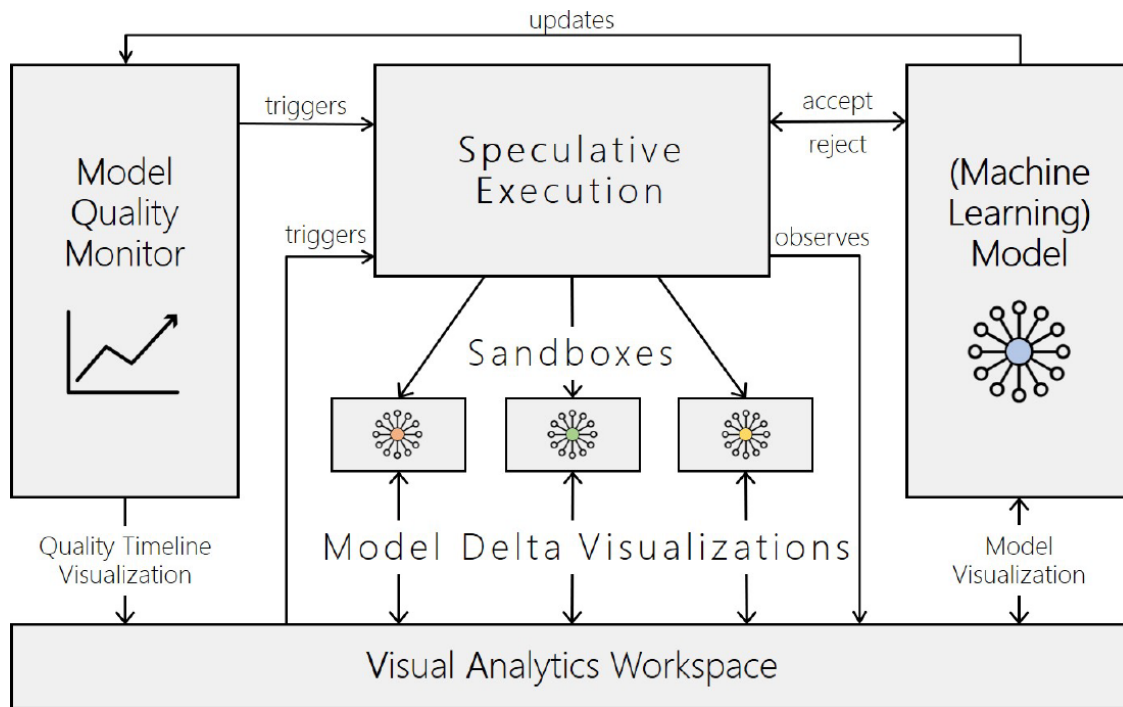


Figure 10.4: The Speculative Execution concept from Sperrle et al [283]. Included under Fair Use, 2019.

mapping for the effects of the interaction was different than the results show. A visual metaphor a system can use to guide interactions that may reduce this confusion is providing a preview of the resulting changes to the visualization after the interaction. Such previews help the analyst decide if the given interaction will produce desired results. The ability to undo an action is also critical. The Speculative Execution model proposed by Sperrle et al. [283] presents another solution to this challenge, computing potential future model states and presenting them in comparison to the current system state. This permits analysts and domain experts to see the effects of interactions before they are initiated (see Figure 10.4).

With respect to the tools described in Chapters 6–8, there is currently no guidance presented to an analyst for future investigative paths. Instead, the tools are designed to permit any form of exploratory data analysis. However, state of the weight vector is tracked throughout an analyst’s exploration, and determining a subset of the dimensions that are most important

to the current exploration strategy is straightforward. Combining this with the Speculative Execution model would allow the systems to precompute a set of potential interactions and identify their effects on the weight vector, thereby presenting suggestions for future interactions to the analyst. Any preattentive technique could be used to communicate these interactions to the analyst, such as blinking the source observation and target cluster.

10.2.2 Analyst and System Understandings of Interactions

When an analyst misunderstands the interactions available to them, they can become frustrated, misuse the system, follow bad exploration paths, or reach incorrect conclusions. Therefore, it is important to understand **How can we mitigate analysts' misunderstandings of the available interactions and their effects?** A proposed solution is to generate standardized interaction terminology to provide the analyst with an understandable mapping between each interaction and the system's reaction to performing that interaction.

However, to provide such terminology, it is important to know **How and when do analyst misunderstandings of interactions occur?** Using a simple menu system as an example, misunderstandings and frustration can occur if a menu item is poorly labeled, leaving the analyst to guess or misinterpret what the menu item does. If instead the menu is well-designed and follows established conventions, like "Save" always being under a menu category called "File," analysts can easily navigate and use the menu system effectively.

Once how and when analysts' misunderstandings of interactions can occur are better understood, this knowledge provides some insight into the analysts' mental mapping of interactions to system responses and updates. Therefore, **How should the system provide an interaction (i.e., how should an interaction be designed) such that the analyst learns how the system maps the interaction to its influence on the visualization?**

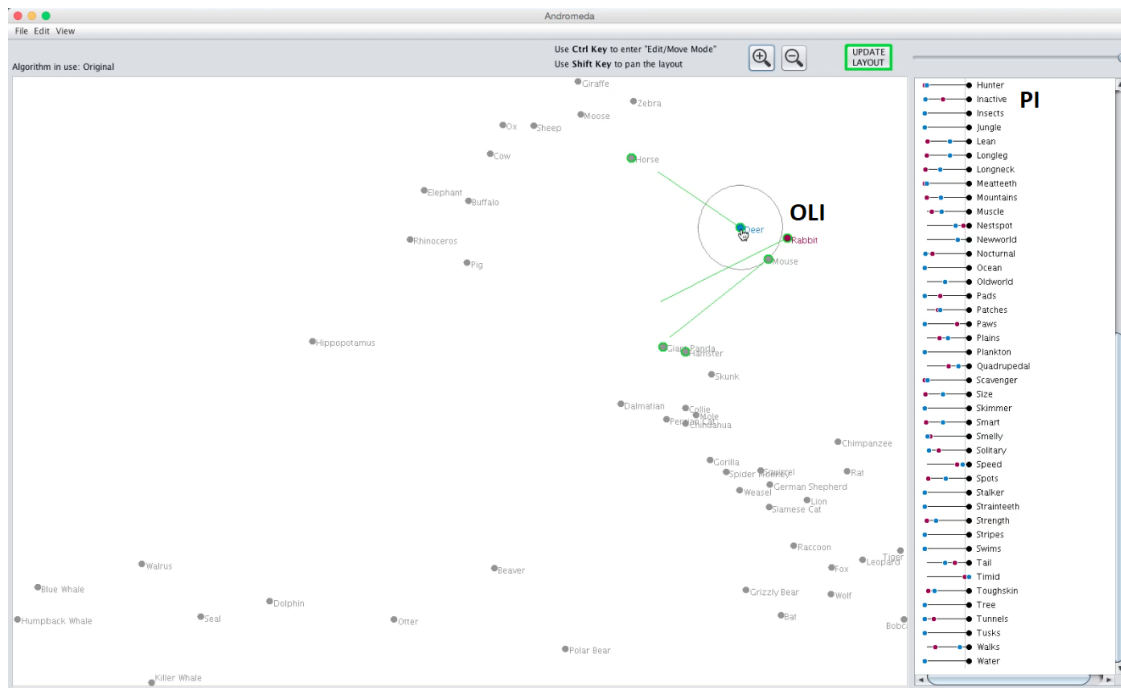


Figure 10.5: Andromeda supports parametric interaction (PI) with interactive slider widgets and observation-level interaction (OLI) with direct manipulations in the projection. Included under Fair Use, 2019.

For example, in Andromeda [274] (Fig. 10.5), parametric interactions (PI) are mapped closely to model parameters. This mapping is expressed to the analyst by displaying and manipulating attribute weights with interactive sliders, an intuitive interaction familiar to analysts. In contrast, Andromeda maps observation-level interactions (OLI) to direct manipulations of the projected observations. Ideally, the analyst's understanding of this interaction should be that such direct manipulations express desired similarity and dissimilarity relationships between observations. However, this mental mapping can be more difficult for analysts to grasp, in part because the mapping between the interaction and model parameters is less clear in OLI. However, another reason may simply be that analysts are not as accustomed to directly interacting with projections of data as they are with sliders. In other words, analysts perceive sliders as having a higher interaction affordance than projected observations.

Though we have not yet performed a formal user study on Gemini, Castor, and Pollux, preliminary results from demonstrations have shown that the interactions supported by these three tools are straightforward. Each of the systems learn new information about the intent of the analyst when an observation is reclassified into a new cluster. However, the default implementation of these systems does not permit analysts to define that only moving an observation out of a cluster (or into a cluster) is the important component; instead, the systems learn from both of these actions to infer why the observation did not belong in the source cluster and why it does belong in the target cluster. Adding a double-click interaction at the beginning or end of the drag action (as proposed in Section 5.5.5) can help to resolve this challenge. Similar techniques could be adopted to enable an analyst to communicate the importance of the precise positioning of an observation within the cluster, permitting each system to learn more about the relative similarities between observations.

10.3 Capture Logs

Interactions performed by an analyst indicate their current intent or goals in their analysis. Logging these task-oriented interactions can help the system disambiguate these intents and goals to further assist the analyst in achieving said goals. In this section, we discuss the challenges of deciding what to log, provenance of logs, and the importance of context. These logs are used in the next phases to update models and personalize visualizations.

10.3.1 The Art of Logging

Ultimately, logs are how the machine is able to capture the progress of the analyst and forms the foundation for understanding analyst intent. Therefore, ensuring that logs are

being captured in a manner that enables the personalization (accomplished in the next step) based on this learned knowledge is important. A critical question to ask is: **How can we detect what the important interactions are?** Logging every pixel of mouse movement and timing every fractional pause is certainly an unsustainable scale of data but guarantees interaction coverage. In contrast, determining that an interaction happened by waiting for a purposeful action such as a mouse click or keystroke may miss important contextual information for that action.

A structured means of considering this challenge, which still has inherent open questions, is to determine: **What to log, when to log, and How to log?** “What to log” focuses on the type of interactions being performed (e.g., clicks and keystrokes); not every interaction may be important (e.g., hovering to see a tooltip). In contrast, “when to log” refers to when such logs should be generated (e.g., only tracking the first click on an object rather than all clicks). These two considerations relate to a notion of scalability of the logs; capturing every interaction means that the logs will become difficult to analyze efficiently or effectively. Lastly, “how to log” centers on media to gather or generate logs, such as eye tracking or audio recording. These considerations dictate what information the system ultimately is able to use in the personalization step and, by extension, how the information can be used. However, a single interaction does not convey much information; a sequence of interactions can convey much more about the intent of the analyst and their current goal. Therefore, being able to track interactions over time is another important consideration. This points to a notion of provenance of interactions and maintaining provenance in the logs [95, 249], and lead to another open question: **How can a generic system retrieve high-level interactions from a sequence of lower-level interactions?** For example, the system can use provenance data to learn which interactions are important or preferred. With this information, the system can keep more detailed logs about the important interactions [149],

which can help dictate what can be or should be learned from such interactions.

For the dimension reduction and clustering tools in this dissertation, including additional logging could be used to support a greater set of semantic interactions. For example, each of these systems currently only processes model updates following cluster reassignment interactions, certainly a low-level interaction. By improving the logging power of the system to track interactions such as hovering for details on demand, repositioning observations within clusters, and even eye tracking where the analyst is currently focusing, we could build a richer set of interactions that could improve the accuracy of weight vector updates.

10.3.2 Contextualizing the Interaction

The idea of provenance leads to considering context in the interaction. **What is the analyst interacting with? What data is currently being used or considered? What is the analyst's current state in their analysis process?** These different facets of context help situate the interaction within the analyst's analysis process. Thus, context can enable the system to personalize the visualization in the next update based on the analyst's current process as opposed to only the analyst's current state. This information about the analyst's process may include information such as what subset of data is most relevant to the analyst and the scope of the analyst's current process. In this sense, logging (particularly with context) opens the "black box" of the analyst to help the system better understand the analyst's analysis process and goals [243].

To assist with identifying context, the interactions can be taxonomized based on how or when they are used and what data they are being used with. Such a taxonomy would help identify what contextual information should be captured alongside the interactions. If a notion of context for an interaction can be predefined, then the system can also begin deter-

mining what the analyst's intent behind using that interaction may be. Existing interaction taxonomies [11, 278, 306, 334] would benefit from contextual extensions.

The limited set of interactions that are currently supported by Gemini, Castor, and Pollux negate the need for an interaction taxonomy for these systems. However, adding additional interactions increases the complexity of the system, increasing the need for a clear and detailed mapping of interaction to its effect on the system. An elicitation study similar to that proposed in Section 5.5.5 can be used to collect and refine additional interactions that can be included in these systems.

10.4 Personalization

With the interaction logs, the system can attempt to infer the analyst's intent and provide an updated, personalized visualization to help with the analyst's analysis. This personalization can focus on analyst characteristics, like personality and experience level, as well as their intent. However, such personalization can be difficult to achieve depending on the information that is logged and how indicative it is towards these personalization goals. The system can communicate this personalized new state back to the analyst, which will be described in the next section.

10.4.1 Predicting Analyst Intent

Predicting the intent of the analyst will help the system provide a visualization that is tailored to the analyst's goals, but predicting their intent is complicated. The analyst may also have multiple, parallel analytical goals when exploring their data. Even if the analyst has a single or primary goal, they will likely perform multiple interactions to achieve that

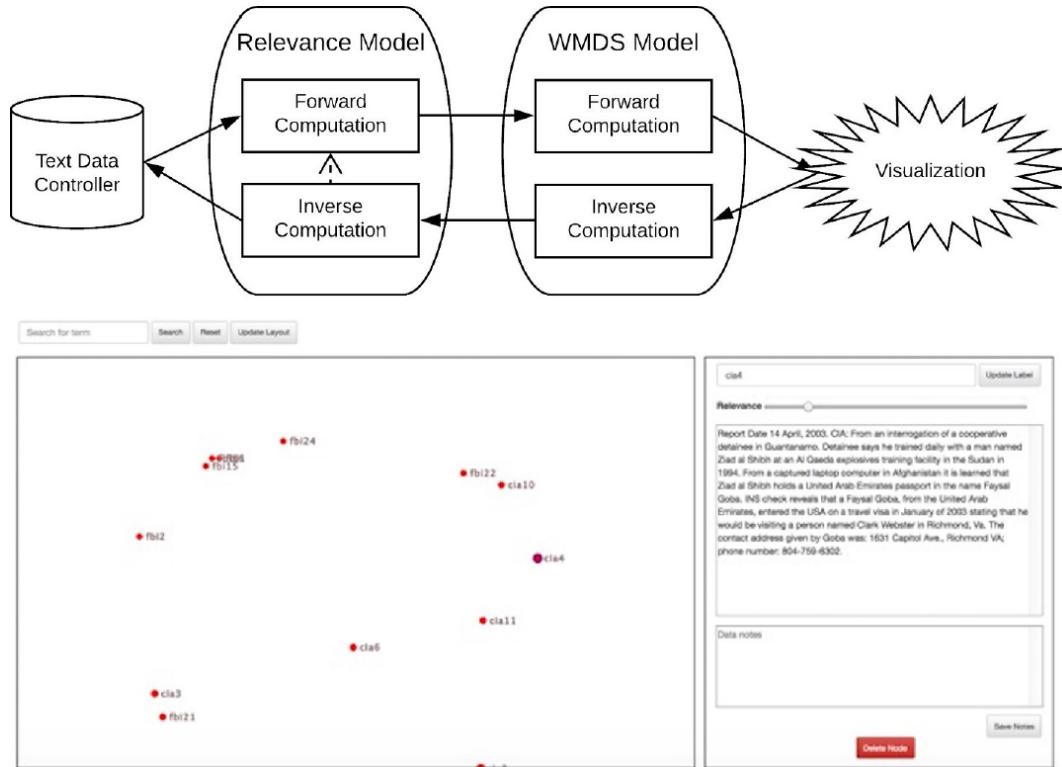


Figure 10.6: An example computational pipeline and system from Dowling et al. [97], in which interactions with document relevance are handled by the Relevance Model and interactions with document positioning are handled by the WMDS Model. Included under Fair Use, 2019.

goal. **How can a system deal with this cardinality issue, appropriately mapping interaction sequences to the goals of the analyst?** In one proposed solution, the Semantic Interaction pipeline proposed by Dowling et al. [97] maps specific subsets of interactions to individual models within a computational pipeline (see Figure 10.6 for an example for the Cosmos system [99]), providing a structure to begin mapping analyst interactions to overarching goals. If the system has advance knowledge of which interactions denote which analyst intents, it use this taxonomy or group related interactions to assist in learning these intents. Additionally, the system can use a series of interactions to gain more information about the analyst’s intent, making techniques such as human-in-the-loop analytics particularly powerful since such techniques require regular feedback from the analyst.

Semantic interaction takes this idea a step further. When performing semantic interactions, the analyst is shielded from the details of the underlying models, but as a result, it may be less clear how their interactions influence the model and thereby influence future visualizations. Though a number of systems that incorporate semantic interactions have been implemented, it is still unclear **What is the optimal means of translating semantic interactions into model updates?** These systems run the gamut from making purely heuristic updates to solving equations to determine precise parameter updates. Understanding the underlying cognitive state of the analyst can give some clues as to the intent of the analyst [2].

Gemini, Castor, and Pollux certainly are not utilizing the optimal means to translate semantic interactions into model updates. Instead, a heuristic learning method is used in order to maintain the real-time interactions within the visualization. However, more rigorous statistical and mathematical modules could be inserted into the code base to make weight updates ideal rather than gradual.

10.4.2 Personalization for Personality

An alternative perspective on personalization is to utilize the analyst's personality traits. These personality traits can help further refine the system's response to analyst interactions to provide an improved visualization and overall experience. For example, if the system knows that the analyst is confused or frustrated [237], perhaps it can guide the analyst toward a useful interaction or relevant data. An example of such a suggested interaction is demonstrated in Fig. 10.7. However, frustration is a short-term trait that can change with time; other traits like expertise are longer-term. Knowing which traits are long-term vs. short-term, **How can a system learn these traits for an individual analyst?** Incorporating such knowledge in the system's personalization is an active challenge and open

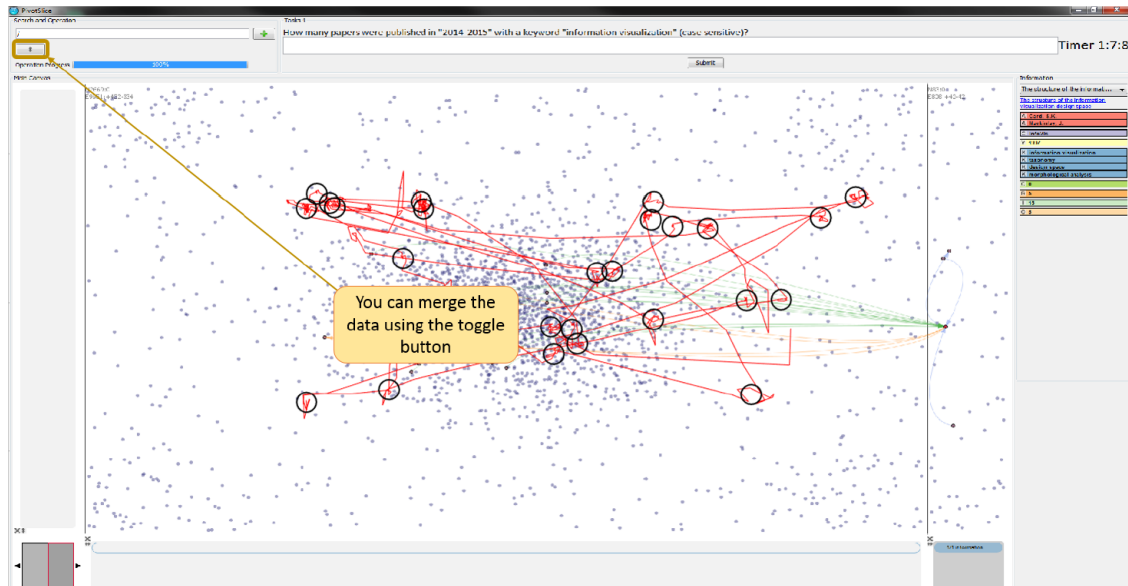


Figure 10.7: After a system detects frustration, it can display suggested interactions [237]. Included under Fair Use, 2019.

research direction. Additionally, **How can analyst personality characteristics best be used to create better and more useful visualizations and analyst experiences?**

Further, personality traits may be characterized by different types of analysts [146, 235, 271, 342]. For example, the system could leverage a grammar of interactions for model tuning to assist expert analysts (e.g., model builders) in their goals. This grammar may be different for other types of analysts, such as domain experts or managers.

This area of research for interactive visualization systems is still quite novel. Further, some techniques to detect individualized information about a user (such as the frustration detection system) are invasive, recording heart rate monitoring, galvanic skin conductance, and eye tracking information. Additional technological refinements are necessary to support actively learning information about these facets of analyst behavior. Until then, personality profiles from the psychology field could profile some starting information about the analyst – their need for control, how willing they are to accept system help, etc.

The Gemini, Castor, and Pollux systems currently treat each analyst as identical, with no personalization profile information incorporated into the interaction/effect mapping. Determining how best to acquire and incorporate an analyst profile into these systems is a broad and open question that requires substantially more research.

10.5 Explainable AI

After the system has updated and personalized the underlying models, the system should now provide an updated visualization and corresponding explanation of the current state of underlying models. Ideally, the system providing this feedback will enhance analyst trust by opening the “black box” of machine learning, thereby enabling the analyst to understand how the system reached this state. In this section, we discuss the difficulty of providing understandable explanations to the analyst. Once the analyst has received this feedback, they can go on to properly evaluate these changes and their suitability in their current analysis process. The communication from the system and the interpretation from the analyst is a tightly-coupled and iterative process, similar to that between the Interactive Visualization and Capture Logs phases (see Fig. 10.2).

10.5.1 Providing Feedback to the Analyst

This step of the loop focuses on how to open the “black box” of the underlying machine learning algorithms to help the analyst better understand the updates that the system has performed. Thus, this step is equivalent to explainable artificial intelligence (XAI). Given the richness of this area of research [78, 94, 136, 222, 267], we wish to simply focus on components of XAI that are relevant to the analyst’s perspective on this step of the process. In

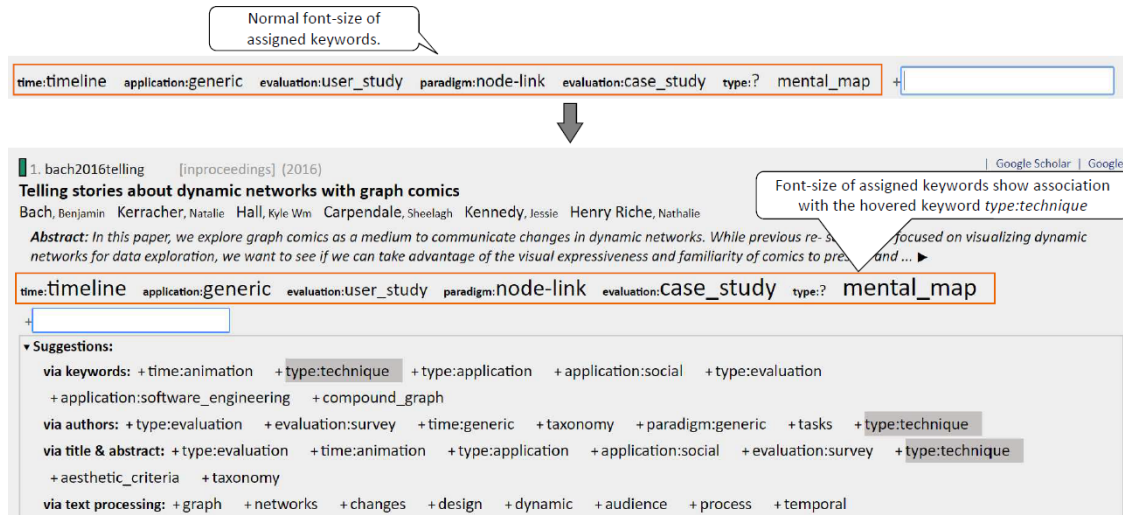


Figure 10.8: The relationship between suggested and assigned keywords encoded in font size. Included under Fair Use, 2019.

particular, an open question is **What types of feedback will benefit the analyst most?** For example, feedback can be given in terms of how specific model parameters changed (requiring expert knowledge of the model itself), or feedback can be provided in more natural and intuitive manners to the analyst (which may obscure details regarding how specific model parameters have changed). As a result, two major considerations in how to provide this feedback to the analyst are (1) how simple the explanations should be and (2) how much background knowledge is required to understand the explanations. For example, the literature tagging application from Agarwal et al. [4] demonstrates a straightforward visual explanation of relationships between suggested and assigned keywords through font size and hovering (Figure 10.8).

The feedback that is provided by the systems implemented in this work provide limited feedback to the analyst regarding the model state. Only the weight vector is displayed onscreen with the visualization, but updates to that weight vector are difficult to track. Additional information could be provided to the analyst to mine information about changes to the weight vector (e.g., via an extra visualization panel that shows changes to a selected

weight over time) or to the current state (e.g., using a visual widget such as the sliders in Andromeda [272]).

The Gemini, Castor, and Pollux systems include several features to communicate information about the visualization and model state back to the analyst. When an observation is assigned to a new cluster in response to newly-learned information, several different visual cues are used to indicate the change (animation for Pollux, node color change for Gemini, node color change and a line back to the source cluster for Castor). Each of these visual cues can be incorporated into all of the systems. The weight vector shown to the right of each of the tool interfaces directly communicates the model state to the analyst; however, adding additional controls to sort the dimensions and additional visual cues such as the slider bars from Andromeda [272] can make this information more tractable to an analyst.

10.5.2 Interpretability and Uncertainty

Related to these major considerations are two other facets of XAI: interpretability and uncertainty. These facets of XAI focus on not just how the analyst can understand the high-level changes that occurred in the underlying models but also the implications and accuracy of these changes. For example, in many dimension reduction techniques, several similar projections may be produced with the same parameters. If the analyst can understand this uncertainty in the projection, perhaps they can better understand more nuanced relationships in the data, thereby creating a more detailed mental model of the data that further aids their analysis process. However, displaying information regarding uncertainty in a manner that is understandable or interpretable to the analyst is a challenging task. As such, a number of uncertainty visualization mechanisms have been designed, including the uncertainty ribbons created by Sanyal et al [268] (Fig 10.9). Indeed, **How do analysts even interpret**

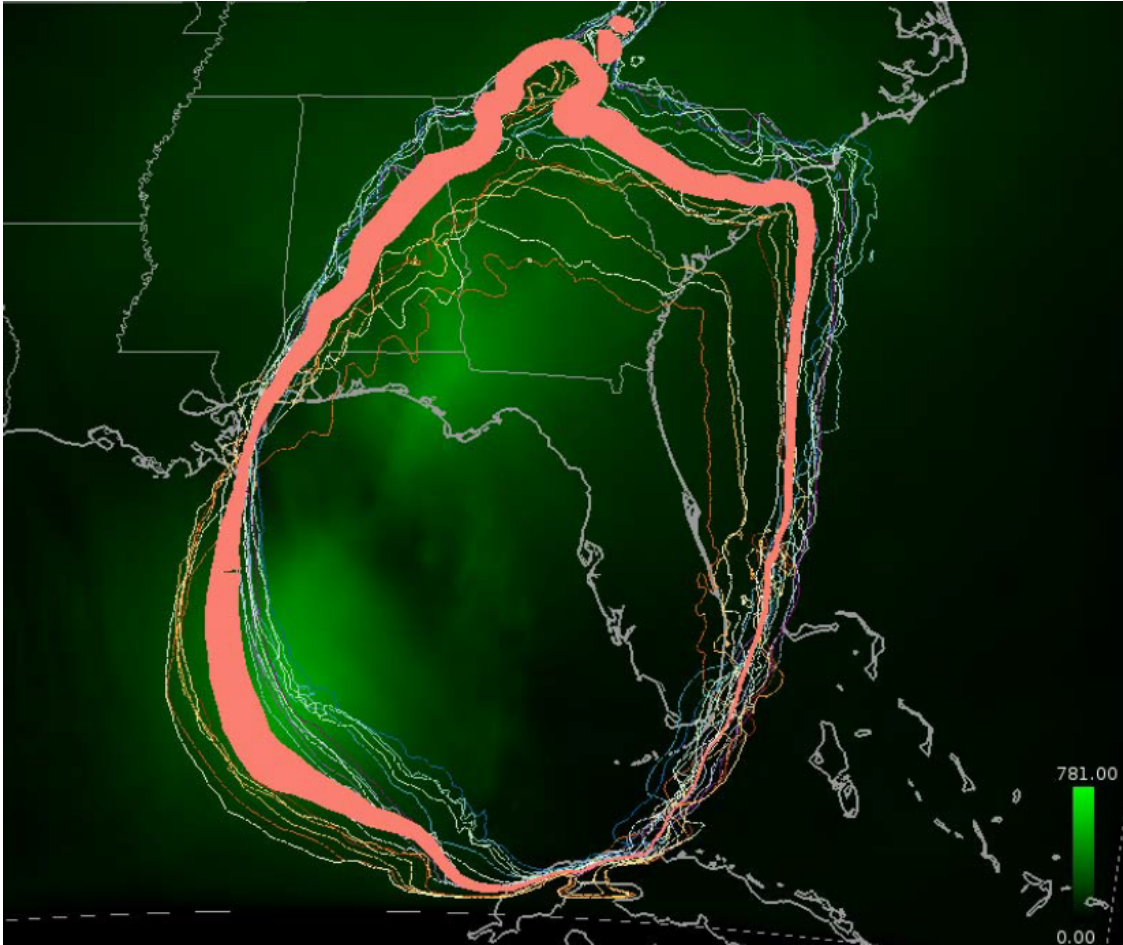


Figure 10.9: An example computational pipeline and system from Dowling et al. [97], in which interactions with document relevance are handled by the Relevance Model and interactions with document positioning are handled by the WMDS Model. Included under Fair Use, 2019.

uncertainty? A system that communicates uncertainty in its output could be interpreted on a scale that ranges between open and honest with its current knowledge and useless because it will not provide a precise answer. Similar ideas centered on the interpretability of the machine learning feedback are also a growing area of research [68, 172, 287].

Gemini, Castor, and Pollux do not currently incorporate uncertainty information, but adding additional visual cues to include this information is relatively straightforward. Additional views that present projection stress graphs, cluster quality graphs, and plots of dimension

weights over time (as noted in Section 7.4) can provide feedback to analysts regarding the quality of their visualization after each interaction. Similarly, including some color-coding information on the edges and nodes can display information about which portions of the visualization fit well and which are not ideally-positioned. Animation could also be used, wiggling nodes that are under high stress and not well-positioned, while nodes that fit well into the projection can remain still. Finding the ideal way to communicate this information to the analyst requires further study.

10.6 User-in-the-Loop Evaluation

After the analyst has received feedback from the system regarding the updated models, they can begin to evaluate the updated visualization and its suitability in their current analysis process. This evaluation process is tightly coupled with the feedback provided by the system and requires one or more metrics, which may be chosen by either the system or the analyst. Choosing the correct metrics and how to integrate them into human-in-the-loop systems is a challenge; these metrics are ultimately determined by the analyst based on their current goal, which may be difficult to externalize in an explicit or easily measurable manner.

10.6.1 How to Evaluate

The goal of evaluating the system is for the analyst to determine the suitability of the updated visualization in their current analysis process. **How should an analyst properly evaluate the solutions that are presented by the system?** This evaluation will be influenced by the feedback provided to the analyst in the previous step; however, this evaluation will be based on user-defined metrics from their own mental model for the visualization's

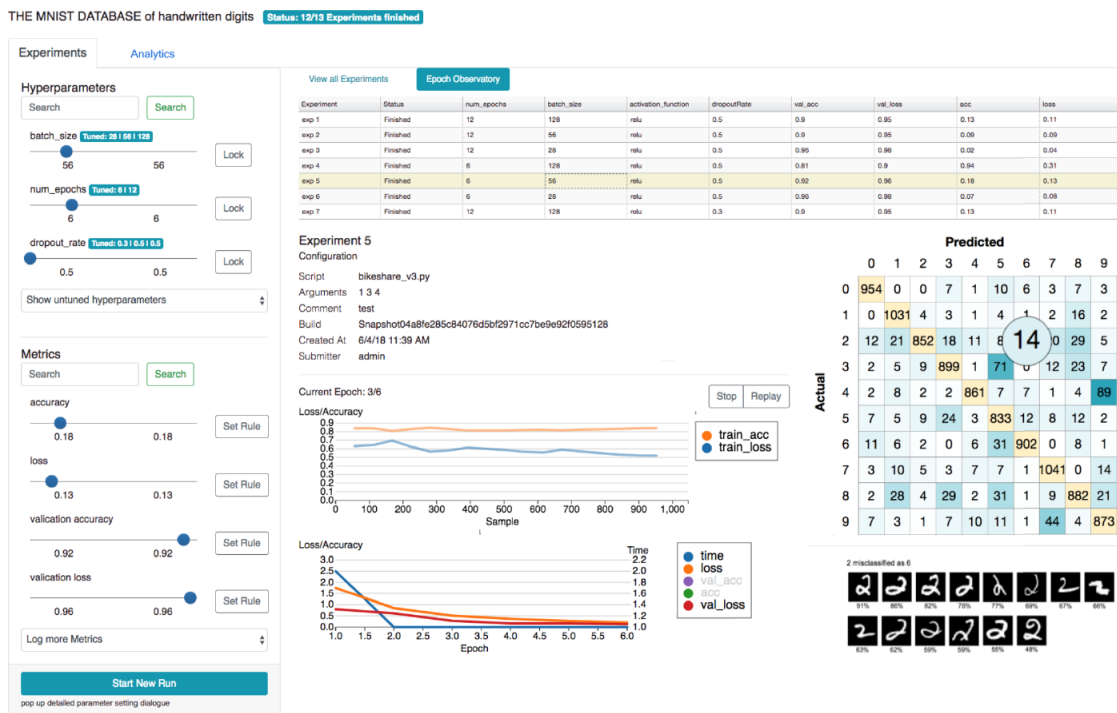


Figure 10.10: The experiment dashboard in HyperTuner [203], allowing an analyst to explore the hyperparameter space. Included under Fair Use, 2019.

suitability. To support this evaluation, the system should be designed to provide the evaluation metrics that are more informative to the analysts. For example, if the analyst's various tasks mandate fast responses from the underlying model, then the system could be built to provide a measurement of how much performance could increase based on which subsets of the data are used in the subsequent calculation (i.e., after the next interaction). For example, HyperTuner [203] gives analysts the ability to perform a hyperparameter search interactively, evaluating a collection of models in a sequence of experiments and passing the results of predefined metrics back to the analyst (see Fig. 10.10). Further, Corbett et al. [76] present a set of ten heuristics for evaluating interactive machine learning systems.

In a similar vein, **How should the analyst and the system handle bias?** If the analyst's higher-level tasks or goals are concerned with bias, the system may be built to provide metrics of bias as part of its feedback. Alternatively, it may learn (through analyst

interaction) that bias is important to the analyst in their current evaluation and therefore *learns* to provide metrics for bias. Similarly, analysts can provide information on past models that they evaluated to be useful. Therefore, the system can learn about its visualization's suitability in the analyst's current analysis process through analyst interactions in the next step (and subsequently logging and personalization).

No precise metrics for the evaluation of the current projection and clustering are currently included in Gemini, Castor, and Pollux, requiring analysts to use their own knowledge of the data to determine whether or not the current visualization appears to make sense. Much like in the previous section, adding additional views that communicate stress and cluster quality can begin to address those visualization quality concerns. Corbett's heuristic evaluation set [76] can provide further guidelines for best designing these evaluation metrics.

10.6.2 Trust

Having a proper evaluation of the system may aid the system in establishing trust with the analyst. But **How does an analyst decide to trust what they see?** How the analyst evaluates the updated visualization is reliant on how much the analyst trusts the update and associated feedback from the previous step. To help the analyst trust the system more, the system can learn what an analyst needs to trust the visualization and feedback. Similarly, it is important for the system to know or learn how and when breakdowns of trust occur to further guide how it provides feedback to the analyst or personalizes visualizations. **Can a system learn what analysts need in order to trust a system?** In other words, trust may change over time, meaning the system may have to re-evaluate the analyst's trust as the system progresses from one state to the next. This is a complex and open research direction.

Gemini, Castor, and Pollux currently make no effort to verify that analysts trust the current

visualization, but trust could be inferred by tracking the number of undo actions performed, or by using eye tracking software to detect analyst frustration [237].

10.7 Self-Correction: Overcoming Incorrect Inferences

A concept that spans multiple steps in the process is the idea of a system self-correcting or overcoming incorrect inferences. This concept targets the fact that trying to learn analyst intent (as is done in the Personalization step) may lead to incorrect conclusions. The result of such incorrect conclusions is that the analyst will be provided with an updated visualization that they evaluate as not being suitable in their current analysis process. The system should find ways to correct for these errors.

A method for this correction that we have already discussed is to continue to try learning the intent from analysts based on their interaction that are subsequently logged. However, how different might the updated visualization be if the system assumed that it was likely wrong in trying to infer the analyst's intent? When the system learns that it was, in fact, wrong, does it understand why its models failed to then know how to correct for this error? For example, **how does the system compensate for the fact that it can understand relationships in the data like correlation but analysts can understand other relationships as well, such as causal relationships?** From a slightly different perspective, what is the difference between a system knowing how to self-correct and an analyst telling it to correct (through interaction)?

The optimal method to perform this correction is largely based on the specific analyst in question. For example, expert analysts are more likely to recognize and understand when the system is providing a nonsensical visualization, whereas novice analysts may not. Similarly, expert analysts can give more detailed feedback such as what the boundaries of the models

should be and what the edge cases are. Therefore, it is important for the system understand the type of analyst and their personality as well as the data itself and what subsets of the data the analyst is interested in.

While it is certainly important to look at examples of systems that have a notion of self-correction, it is also useful to look at examples of systems that do this poorly. A frequently-cited example is Microsoft's Clippy. Clippy has become the Platonic ideal of what an AI should *not* be. The graphic popped up and interrupted the analyst frequently, many times with incorrect assumptions about what the analyst was currently trying to accomplish. Furthermore, it never retained knowledge of whether it was right or wrong in trying to learn the analyst's intent, meaning Clippy never tried to learn analyst intent differently or did any self-correction.

Gemini, Castor, and Pollux permit analysts to “undo” an interaction simply by performing the opposite interaction: move an observation from Cluster B to Cluster A in order to negate the previous interaction from Cluster A to Cluster B. However, this becomes more difficult when an analyst realized that their mistake was several interactions in the past. Including evaluation and uncertainty visual cues in the visualization can help analysts to uncover these issues more quickly. Additionally, the system itself could also monitor evaluation metrics, directly alerting the analyst whenever they begin to force interactions that are not necessarily supported by the data.

10.8 Discussion

The ideas presented here are a reflection of the research and discussion that happened during the IEEE VIS 2018 Workshop on Machine Learning from User Interaction for Visualization and Analytics. As opposed to a thorough literature exploration of the phases in the human-

in-the-loop process (Fig. 10.2), we instead present an overview of the common themes and ideas that arose from the workshop. Additionally, we identify and discuss current research questions and considerations of researchers at the workshop. Thus, a thorough survey of interactive machine learning, human-in-the-loop analytic systems, and human-centered machine learning literature could uncover issues and open questions is not discussed in this work, and we anticipate that the future research directions in these areas will be much richer than the set of ideas presented here.

10.9 Conclusion

Machine learning from user interaction can offer many new opportunities for visualization and analytics, as demonstrated by the number of open research questions. As such, much work remains to be accomplished in this space, and initial research successes demonstrate encouraging results. Overall, the workshop discussants came to the clear conclusion that the visualization research community should continue with future workshops and other publication, outreach, education, funding, and community building initiatives for this topic. As demonstrated through the discussion points at the end of each subsection, existing tools can also be extended to support this research agenda and further explore the open questions presented in this chapter.

Chapter 11

Discussion and Conclusion

11.1 Discussion

The Discussion section revisits six topics that were addressed in various places through the previous chapters. The goal in the subsections below is to synthesize these topics into concise lessons identified from performing this research.

11.1.1 How Different Are These Operations?

Chapter 6 briefly proposes the idea that clustering and dimension reduction are not substantially different operations. The chapters notes that “in a way making cluster assignments is equivalent to 1D dimension reduction, noting that the cluster assignment is the primary dimension of organization.” The tight relationship between these two operations was also seen in the survey of dimension reduction tools in Section 2.2.6, and was also hinted at in Chapter 9 when study participants occasionally struggled to define their interactions as purely spatial or grouping. This all leads to the question: “Is clustering really just discretized dimension reduction?”

In true computer science fashion, the answer uncovered within this research boils down to “it depends.” Spatial and grouping operations do appear to be different cognitive actions, but are still tightly related. In the study performed in Chapter 9, we often needed to

clarify whether two observations that were positioned close together were being considered a cluster, or if they were just two observations that happened to be close together. Often, the participants themselves were not certain which action they were performing, only responding to the question after giving their response considerable thought. The knowledge that clusters are subjective structures that are often defined by the analyst [118] makes this determination an even greater challenge to resolve.

This relationship between dimension reduction and clustering is also supported mathematically in specific instances. As we noted in our review of the literature, Ding and He [89] proved that principal components are the continuous solutions to the discrete cluster membership indicators for k -means clustering, indicating that Principal Component Analysis dimension reduction implicitly performs clustering as well. Topic modeling algorithms also effectively perform both dimension reduction and clustering, creating groups of documents (clusters) that are defined by a small set of terms (a lower-dimensional description of a document than the term frequency counts).

11.1.2 “With Respect to What”

The “With Respect to What” problem detailed by Self et al. [273] described a direct manipulation interaction challenge with projections. In general, if an analyst repositions an observation within the projection, the interaction doesn’t have meaning without judging what that observation has been moved with respect to. Choosing from all of the possibilities is a difficult problem without providing some additional visual cues for how the interaction will be interpreted by the system, as well as some additional interactions that can permit an analyst to disambiguate the interaction from several possibilities.

The interaction discussions in Chapter 5 address the “With Respect to What” problem when clusters have been introduced. In a way, introducing clustering simplifies this challenge by providing a smaller set of potential interaction targets. However, this is only true if clusters are the only permitted interaction targets. Introducing clusters can also increase the complexity of interpreting the interaction. For example, an incomplete list of the interactions that might be intended in the motivating example in Section 5.2 and Figure 5.4 include:

- Move the Grizzly Bear into the Predators cluster
- Move the Grizzly Bear out of the Pets cluster
- Move the Grizzly Bear closer to the Leopard
- Move the Grizzly Bear closer to the Leopard, Wolf, and Lion
- Move the Grizzly Bear away from the German Shepherd
- Move the Grizzly Bear away from the dogs
- Move the Grizzly Bear into the Predators cluster, but closer to the Lion and Leopard than the Fox
- Move the Grizzly Bear into the Predators cluster and out of the Pets cluster, while also not changing its relationship to the Large Herbivorous Animals cluster

Disambiguating from all of these possible analyst intents and more presents a significant challenge to the design of clear and simple interactions. The work included in this dissertation makes no claim of solving this interaction challenge, but instead attempts to better enumerate the complexities involved when incorporating dimension reduction and clustering together.

11.1.3 The HCI and ML Perspectives

This dissertation occasionally makes use of terminology from the machine learning community as well as from the human-computer interaction community. This choice is not

accidental or unintentional, as the research contained within this dissertation exists at the intersection of both fields: we include discussions of both the semi-supervised training of machine learning algorithms and design and interaction considerations for interactive visualization tools. Perhaps the clearest example of such symmetry within this work comes from the overlapping ideas of inferring the intent of a user and mapping that intent to a learned metric in the system.

From the HCI perspective, this dissertation contributes discussions of design considerations, methods of responding to interaction ambiguity, and example interactive interfaces. For ML, this dissertation contains algorithmic descriptions, several implementations of interactive systems that can learn from analyst interactions, and discussions of learning issues within this design space. Both of these perspectives address separate but concurring facets of the same problem.

11.1.4 Design Lessons

When combining dimension reduction and clustering into the same system, there are a number of design decisions to be made. This is particularly true simply because of the immense design space of algorithms, tasks, visualizations, and interactions that can be supported by these algorithms.

Chapter 4 noted four overarching design decisions to be addressed when creating the visualization. Considering the tasks that a system should support, a designer should determine what properties of the data the visualization needs to highlight, as well as which properties the system and analyst are working cooperatively to discover. The tasks that the system must address also impacts the choice of computational pipeline when generating a visualization from the provided data, as well as how the individual models should learn from analyst

feedback (e.g., a single distance function?). This chapter also addresses the complexity of encoding distance and cluster membership information when both dimension reduction and clustering algorithms are present in a system, and notes that designers should consider how to create a visualization that still operates when these algorithms are in conflict or are generally competing. A set of design decisions are specifically noted in Table 4.2.

Chapter 5 focuses on the interactions that should be supported by such interactive visualization systems, and picks out a number of dimensions that should be considered when creating interactions. For example, is the target of the interaction an observation, a cluster, or both? Should the interaction be applied to the nearest observation, or the nearest n observations, or all observations within a cluster? Or all observations in the projection? Other factors uncovered include the importance of the source or target, the level at which the analyst is thinking, and the influence of the visual design. Section 5.5.4 describes factors to consider when interpreting the intent of an interaction.

Chapter 7 discusses ways by which the clustering-first technique of Pollux could be extended in design. These include the selection of edge classes and the corresponding effect on performance, the selection of edge class weights, alternate visual representations, providing the analyst with control of the number of clusters, supporting hierarchical clustering, and identifying when to learn from an analyst. Each of these methods is expanded upon in Section 7.2.

Finally, Chapter 8 also includes a set of design lessons that were revisited from previous chapters during the evaluation of Gemini, Castor, and Pollux in Section 8.4.2. Chapter 9 further notes several design lessons uncovered from the cognitive study, detailed in Table 9.2.

Considering the design lessons from the individual topics throughout this dissertation can combine to create design lessons for entire tools, a case of the whole being greater than the sum of its parts. When asking for a global “best” solution for designing an interactive

dimension reduction and clustering system, the only possible answer to provide is “it depends.” However, providing additional information by, for example, supplying the tasks that must be supported by the system can begin to generate design recommendations to refine other details of the system (e.g., algorithm selection and order, visual representation, and interactions). The ultimate goal of these lessons is to design a system that helps analysts to understand their dataset by means of the visualization that has been produced.

11.1.5 Generalizing to Other Model Families

In recent years, analysts have worked to explore and draw conclusions from increasingly larger datasets. As a result, visual analytics tools continue to grow more complex, with computational pipelines that transform data into interactive visualizations now often consisting of multiple analytical models. These multi-model systems are becoming prevalent, and include but are not limited to combinations such as relevance and similarity [38], sampling and projection [239], and control point selection and manipulation [214].

There are a large number of computational models in the data science toolbox that can adapt ideas contained in this dissertation to create future interactive visualization pipelines. One might imagine a computational pipeline that transforms data into a visualization through interactive, intelligent filtering, regression, and anomaly detection prior to visualizing the data. Each of these models could be independently trained through interactive analyst-driven feedback that is mapped to specific interactions. A dataset of documents could further benefit from text summarization and term co-occurrence models in such a pipeline.

The chapters in this dissertation detail a number of considerations for system developers who intend to combine two or more models into a computational visualization pipeline. These considerations include:

- Which specific algorithms from the model families will be implemented, what are their properties, and how can they be used? (Chapters 2 and 3)
- What tasks does the system need to support? (Section 4.1)
- How should the algorithms be ordered in the pipeline? (Section 4.2)
- What visual representation is best to communicate the algorithm output? (Section 4.3)
- How should an analyst interact with these algorithms? (Section 5)

11.1.6 Future Interactive Algorithm Opportunities

Sections 2.1 and 3.1 present a wide range of dimension reduction and clustering algorithms that exist in the literature, but only a small subset of them are covered by the systems described in the sections that follow (Sections 2.2 and 3.2). Indeed, the vast majority of interactive visualization literature is focused on techniques like PCA, MDS, LDA, and k -means. The algorithm surveys enumerate a variety of future research directions for interactive visualization tools. Indeed, we identified the need for more interactive subspace clustering in Section 9.5.3 when discussing the complex spaces created by study participants. Moving beyond k -means to techniques such as interactive Dirichlet process clustering [292, 293] could support other clustering assignments within systems like Gemini, Castor, and Pollux.

11.2 Limitations and Future Work

In this section, we summarize limitations to the research surveys provided in early chapters of this dissertation, to the tools created in later chapters, and to the cognitive study presented in Chapter 9. Proposed methods of addresses these limitations in future work are also included in the discussion.

11.2.1 Surveys

The primary limitation of the literature surveys throughout Chapters 2–5 is the focus on literature from statistics and visualization journals and conferences. While this supports the focus of this research, there is also some related work that can be found in conferences such as ICML and KDD. The surveys presented here could be supplemented by research from related fields, such as human-centered machine learning. Additionally, research from HCI literature can contribute to the design of interactions in the tools described in later chapters.

11.2.2 Tools

One of the main limitations of the Gemini, Castor, and Pollux tools is the choice of force-directed layout and k -means clustering for the algorithms. Much like many dimension reduction algorithms, the computational complexity of the force-directed layout limits the scalability of the system to a few hundred observations at best. While some techniques have been created to somewhat overcome this limitation (for example, [80, 164]), there are still limits on the scalability of these techniques which prevent easy exploration of full, large datasets. Instead, techniques just as sampling and selection of representation observations must be employed for such datasets. Likewise, k -means suffers from limitations noted in Section 3.1: best identifying clusters of similar covariance that are convex in shape. While the Lloyd’s algorithm approach to approximating an ideal k -means solution does not cause the same scalability constraints as the dimension reduction algorithm selection, it does limit the accuracy of clusters identified.

The three tools are also currently limited to quantitative datasets. However, extending these tools to support collections of documents or other complex observations is as straightforward as writing a module to convert these complex features into a continuous numerical represen-

tation. Techniques such as Probabilistic Latent Semantic Visualization (PLSV) incorporate a topic model processing step to group documents, followed by a dimension reduction stage to embed the documents into a 2D projection [166]. Clustering could also be introduced into existing tools such as Cosmos [99] and StarSPIRE [38]. We have also not yet performed a user study on these three tools, instead choosing to focus our efforts on understanding the behavior of users more generally when performing spatialization and grouping interactions.

This research further primarily considers dimensionally-reduced projections in which observations are mapped to a point; however, other alternative visual representations exist. For example, parallel coordinate plots can be used to represent the values of observations across either input or synthetic dimensions. The lines that represent each observation can then be bundled or clustered. Similarly, a small glyph can be used to encode information about some of the dimensions selected from an input dataset to provide additional information to an analyst when interacting with a projection.

11.2.3 Study

The study described in Chapter 9 also has limitations, in that we only performed the study on a single dataset using a sample of convenience and with a limited number of participants. Such issues can be remedied by future studies that examine additional participants, varying the datasets supplied and selecting from a broader pool of potential participants.

11.3 Conclusion

This work investigated the systematic combination of dimension reduction and clustering in visual analytics. Chapters 2 and 3 surveyed the existing body of knowledge in the area

of each algorithm, first discussing standard dimension reduction and clustering algorithms, followed by understanding how dimension reduction and clustering are currently used to enhance the sensemaking process in interactive applications. Following this, Chapters 4 and 5 discussed the design space of combining both algorithm families, both in projecting the data and in allowing an analyst to interact with those projections. Chapters 6 and 7 explored two points in this design space, first a tool with dimension reduction preprocessing for clustering, second a tool with clustering preprocessing for dimension reduction. Chapter 9 followed this discussion of the design space with an investigation of how analysts perform dimension reduction and clustering tasks when presented with an unfamiliar dataset.

This work enables analysts to more efficiently explore high-dimensional data through visualization and interaction, enhancing the exploratory data analysis process. Through an incremental formalism process, analysts gain new levels of insight from their exploration of the clusters and the interactions enabled upon them; such insights could not be gained from previous tools that did not include the combination of dimension reduction, clustering, and semantic interaction.

Additional components of the tools generated, such as the list of weights included in the Castor system from Chapter 6, serve as a form of Explainable AI, communicating the current state of underlying models to the analyst. This explainable artificial intelligence is introduced into “human in the loop” design via these interactive projections, putting a human analyst into control of the analysis process rather than simply serving in a validation role. As such, this work explores both sides of the ML/VIS partnership, using VIS to explain ML while simultaneously using ML to support VIS.

This work could be further extended in multiple directions beyond this dissertation. In the first extension at the intersection of visualization and data science, lessons learned from this systematic combination of dimension reduction and clustering algorithms could be ex-

tended to other tools from the Data Science Toolbox, including but not limited to regression, classification, ensemble techniques, and basic and advanced machine learning. In the second extension at the interaction of visualization and user modeling, long-term user behavior such as need for control and short-term user behavior like frustration could be detected and taken into account for these interactive systems.

Bibliography

- [1] James Abello, Frank van Ham, and Neeraj Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, September 2006. ISSN 1077-2626. doi: 10.1109/TVCG.2006.120. URL <http://dx.doi.org/10.1109/TVCG.2006.120>.
- [2] Mohamad Aboufoul, Ryan Wesslen, Isaac Cho, Wenwen Dou, and Samira Shaikh. Using hidden markov models to determine cognitive states of visual analytic users. In *Proceedings of the Machine Learning from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.
- [3] Elke Aichert, Christian Böhm, and Peer Kröger. Deli-clu: boosting robustness, completeness, usability, and efficiency of hierarchical clustering by a closest pair ranking. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 119–128. Springer, 2006.
- [4] Shivam Agarwal, Jürgen Bernard, and Fabian Beck. Computer-supported interactive assignment of keywords for literature collections. In *Proceedings of the Machine Learning from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.
- [5] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. *On the Surprising Behavior of Distance Metrics in High Dimensional Space*, pages 420–434. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. ISBN 978-3-540-44503-6. doi: 10.1007/3-540-44503-X_27. URL http://dx.doi.org/10.1007/3-540-44503-X_27.
- [6] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan.

- Automatic subspace clustering of high dimensional data. *Data Mining and Knowledge Discovery*, 11(1):5–33, Jul 2005. ISSN 1573-756X. doi: 10.1007/s10618-005-1396-1. URL <https://doi.org/10.1007/s10618-005-1396-1>.
- [7] Mark S Aldenderfer and Roger K Blashfield. *Cluster analysis*. SAGE publications, Beverly Hills, USA, 1984.
- [8] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, and M. Gleicher. Serendip: Topic model-driven visual exploration of text corpora. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 173–182, Oct 2014. doi: 10.1109/VAST.2014.7042493.
- [9] B. Alper, N. Riche, G. Ramos, and M. Czerwinski. Design study of linesets, a novel set visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2259–2267, Dec 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2011.186.
- [10] J. Alsakran, Y. Chen, Y. Zhao, J. Yang, and D. Luo. Streamit: Dynamic visualization and interactive exploration of text streams. In *2011 IEEE Pacific Visualization Symposium*, pages 131–138, March 2011. doi: 10.1109/PACIFICVIS.2011.5742382.
- [11] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 111–117, Oct 2005. doi: 10.1109/INFVIS.2005.1532136.
- [12] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney Tan. Effective end-user interaction with machine learning. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.
- [13] Saleema Amershi, James Fogarty, and Daniel Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI*

- Conference on Human Factors in Computing Systems*, CHI '12, pages 21–30, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207680. URL <http://doi.acm.org/10.1145/2207676.2207680>.
- [14] C. Andrews and C. North. Analyst’s workspace: An embodied sensemaking environment for large, high-resolution displays. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 123–131. IEEE, Oct 2012. doi: 10.1109/VAST.2012.6400559.
- [15] Christopher Andrews, Alex Endert, and Chris North. Space to think: Large high-resolution displays for sensemaking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 55–64, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9. doi: 10.1145/1753326.1753336. URL <http://doi.acm.org/10.1145/1753326.1753336>.
- [16] G. Andrienko, N. Andrienko, S. Rinzivillo, M. Nanni, D. Pedreschi, and F. Giannotti. Interactive visual clustering of large collections of trajectories. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 3–10, Oct 2009. doi: 10.1109/VAST.2009.5332584.
- [17] Gennady Andrienko, Natalia Andrienko, Salvatore Rinzivillo, Mirco Nanni, and Dino Pedreschi. A visual analytics toolkit for cluster-based classification of mobility data. In Nikos Mamoulis, Thomas Seidl, Torben Bach Pedersen, Kristian Torp, and Ira Assent, editors, *Advances in Spatial and Temporal Databases*, pages 432–435, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02982-0.
- [18] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, pages 49–60,

- New York, NY, USA, 1999. ACM. ISBN 1-58113-084-8. doi: 10.1145/304182.304187. URL <http://doi.acm.org/10.1145/304182.304187>.
- [19] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press. ISBN 978-0-9749039-5-8. URL <http://dl.acm.org/citation.cfm?id=1795114.1795118>.
- [20] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991. ISSN 0360-0300. doi: 10.1145/116873.116880. URL <http://doi.acm.org/10.1145/116873.116880>.
- [21] Arindam Banerjee, Inderjit Dhillon, Joydeep Ghosh, Srujana Merugu, and Dharmendra S. Modha. A generalized maximum entropy approach to bregman co-clustering and matrix approximation. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 509–514, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1. doi: 10.1145/1014052.1014111. URL <http://doi.acm.org/10.1145/1014052.1014111>.
- [22] Jeffrey D Banfield and Adrian E Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.
- [23] Josh Barnes and Piet Hut. A hierarchical $O(n \log n)$ force-calculation algorithm. *Nature*, 324(6096):446, 1986.
- [24] Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. *Active Semi-Supervision for Pairwise Constrained Clustering*, pages 333–344. Springer, 2004. doi: 10.1137/1.9781611972740.31.

- [25] Sumit Basu, Danyel Fisher, Steven M Drucker, and Hao Lu. Assisting users with clustering tasks by combining metric learning and classification. In *AAAI*, 2010.
- [26] Michael Behrisch, Fatih Korkmaz, Lin Shao, and Tobias Schreck. Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 43–52. IEEE, 2014.
- [27] Richard Ernest Bellman. *Dynamic Programming*. Dover Publications, New York, NY, 2003. ISBN 0486428095.
- [28] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8): 1798–1828, Aug 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.50.
- [29] M.L. Bermingham, R. Pong-Wong, A. Spiliopoulou, C. Hayward, I. Rudan, H. Campbell, A.F. Wright, J.F. Wilson, F. Agakov, P. Navarro, and C.S. Haley. Application of high-dimensional feature selection: Evaluation for genomic prediction in man. *Scientific Reports*, 5, May 2015. doi: 10.1038/srep10312.
- [30] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981. ISBN 0306406713.
- [31] Patrick Billingsley. *Probability and Measure*. John Wiley and Sons, second edition, 1986.
- [32] Christopher M Bishop, Markus Svensén, and Christopher KI Williams. Gtm: The generative topographic mapping. *Neural computation*, 10(1):215–234, 1998.
- [33] David M. Blei and Michael I. Jordan. Variational inference for dirichlet process mixtures. *Bayesian Analysis*, 1:121–144, 2005.

- [34] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [35] Ingwer Borg and Patrick Groenen. Modern multidimensional scaling: Theory and applications. *Journal of Educational Measurement*, 40(3):277–280, 2003.
- [36] Lydia Boudjeloud-Assala, Philippe Pinheiro, Alexandre Blansch, Thomas Tamisier, and Benot Otjacques. Interactive and iterative visual clustering. *Information Visualization*, 15(3):181–197, 2016. doi: 10.1177/1473871615571951.
- [37] Nadia Boukhelifa, Anastasia Bezerianos, and Evelyne Lutton. *Evaluation of Interactive Machine Learning Systems*, pages 341–360. Springer International Publishing, Cham, 2018. ISBN 978-3-319-90403-0. doi: 10.1007/978-3-319-90403-0_17. URL https://doi.org/10.1007/978-3-319-90403-0_17.
- [38] L. Bradel, C. North, L. House, and S. Leman. Multi-model semantic interaction for text analytics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 163–172, Oct 2014. doi: 10.1109/VAST.2014.7042492.
- [39] Lauren Bradel, Nathan Wycoff, Leanna House, and Chris North. Big text visual analytics in sensemaking. In *2015 Big Data Visual Analytics (BDVA)*, pages 1–8. IEEE, 2015.
- [40] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, Dec 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.124.
- [41] Matthew Brehmer, Michael Sedlmair, Stephen Ingram, and Tamara Munzner. Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences. In *Proceedings of the Fifth Workshop on Beyond Time and Errors:*

- Novel Evaluation Methods for Visualization*, BELIV '14, pages 1–8, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3209-5. doi: 10.1145/2669557.2669559. URL <http://doi.acm.org/10.1145/2669557.2669559>.
- [42] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 83–92, Oct 2012. doi: 10.1109/VAST.2012.6400486.
- [43] E. T. Brown, A. Ottley, H. Zhao, Q. Lin, R. Souvenir, A. Endert, and R. Chang. Finding waldo: Learning about users from their interactions. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1663–1672, Dec 2014. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2346575.
- [44] Eli T. Brown, Sriram Yarlagadda, Kristin Cook, Remco Chang, and Alex Endert. Modelspace: Visualizing the trails of data models in visual analytics systems. In *Proceedings of the Machine Learning from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.
- [45] Andreas Buja, Dianne Cook, and Deborah F. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996. doi: 10.1080/10618600.1996.10474696.
- [46] Wolfgang Büschel, Patrick Reipschläger, Ricardo Langner, and Raimund Dachsel. Investigating the use of spatial interaction for 3d data visualization on mobile devices. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, ISS '17, pages 62–71, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4691-7. doi: 10.1145/3132272.3134125. URL <http://doi.acm.org/10.1145/3132272.3134125>.

- [47] W. Cancino, N. Boukhelifa, and E. Lutton. Evographdice: Interactive evolution for visual analytics. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8, June 2012. doi: 10.1109/CEC.2012.6256553.
- [48] Y. Cao and L. Wang. Automatic Selection of t-SNE Perplexity. *arXiv e-prints*, August 2017.
- [49] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.
- [50] J Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. *Psychometrika*, 35(3):283–319, 1970.
- [51] M. Cavallo and . Demiralp. Clustrophile 2: Guided visual clustering analysis. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):267–276, Jan 2019. ISSN 1077-2626. doi: 10.1109/TVCG.2018.2864477.
- [52] Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern recognition*, 28(5):781–793, 1995.
- [53] T. Chandler, M. Cordeil, T. Czauderna, T. Dwyer, J. Glowacki, C. Goncu, M. Klapperstueck, K. Klein, K. Marriott, F. Schreiber, and E. Wilson. Immersive analytics. In *2015 Big Data Visual Analytics (BDVA)*, pages 1–8, Sep. 2015. doi: 10.1109/BDVA.2015.7314296.
- [54] Angelos Chatzimparmpas, Rafael Messias Martins, and Andreas Kerren. t-visne: A visual inspector for the exploration of t-sne. In *IEEE Information Visualization (VIS’18), Berlin, Germany, 21-26 October, 2018*, 2018.

- [55] H. Chen, W. Chen, H. Mei, Z. Liu, K. Zhou, W. Chen, W. Gu, and K. L. Ma. Visual abstraction and exploration of multi-class scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1683–1692, Dec 2014. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2346594.
- [56] X. Chen, J. Z. Self, L. House, J. Wenskovitch, M. Sun, N. Wycoff, J. R. Evia, S. Leman, and C. North. Be the data: Embodied visual analytics. *IEEE Transactions on Learning Technologies*, 11(1):81–95, Jan 2018. ISSN 1939-1382. doi: 10.1109/TLT.2017.2757481.
- [57] Xin Chen, Jessica Zeitz Self, Leanna House, and Chris North. Be the data: A new approach for immersive analytics. In *IEEE Virtual Reality 2016 Workshop on Immersive Analytics*, March 2016.
- [58] Yizong Cheng and George M Church. Biclustering of expression data. In *ISMB*, volume 8, pages 93–103, 2000.
- [59] E. H. Chi, L. Hong, J. Heiser, and S. K. Card. Scentindex: Conceptually reorganizing subject indexes for reading. In *2006 IEEE Symposium On Visual Analytics Science And Technology*, pages 159–166, Oct 2006. doi: 10.1109/VAST.2006.261418.
- [60] Ed H. Chi, Lichan Hong, Michelle Gumbrecht, and Stuart K. Card. Scenthighlights: Highlighting conceptually-related sentences during reading. In *Proceedings of the 10th International Conference on Intelligent User Interfaces, IUI '05*, pages 272–274, New York, NY, USA, 2005. ACM. ISBN 1-58113-894-6. doi: 10.1145/1040830.1040895. URL <http://doi.acm.org/10.1145/1040830.1040895>.
- [61] Ed H Chi, Lichan Hong, Julie Heiser, Stuart K Card, and Michelle Gumbrecht. Scentindex and scenthighlights: productive reading techniques for conceptually reorganizing subject indexes and highlighting passages. *Information Visualization*, 6(1):32–47, 2007.

- [62] J. Choo, S. Bohn, and H. Park. Two-stage framework for visualization of clustered high dimensional data. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 67–74, Oct 2009. doi: 10.1109/VAST.2009.5332629.
- [63] J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, Dec 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.212.
- [64] Jason Chuang and Daniel J Hsu. Human-centered interactive clustering for data analysis. *Conference on Neural Information Processing Systems (NIPS). Workshop on Human-Propelled Machine Learning*, 2014.
- [65] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12*, pages 74–77, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1287-5. doi: 10.1145/2254556.2254572. URL <http://doi.acm.org/10.1145/2254556.2254572>.
- [66] Haeyong Chung and Chris North. Savil: Cross-display visual links for sensemaking in display ecologies. *Personal Ubiquitous Comput.*, 22(2):409–431, April 2018. ISSN 1617-4909. doi: 10.1007/s00779-017-1091-4. URL <https://doi.org/10.1007/s00779-017-1091-4>.
- [67] Haeyong Chung, Chris North, Jessica Zeitz Self, Sharon Chu, and Francis Quek. Visporter: Facilitating information sharing for collaborative sensemaking on multiple displays. *Personal Ubiquitous Comput.*, 18(5):1169–1186, June 2014. ISSN 1617-4909. doi: 10.1007/s00779-013-0727-2. URL <http://dx.doi.org/10.1007/s00779-013-0727-2>.

- [68] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feed-back recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075, 2015.
- [69] Andy Cockburn and Bruce McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 203–210. ACM, 2002. ISBN 1-58113-453-3. doi: 10.1145/503376.503413. URL <http://doi.acm.org/10.1145/503376.503413>.
- [70] Anni Coden, Marina Danilevsky, Daniel Gruhl, Linda Kato, and Meena Nagarajan. A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 237–245. SIAM, 2017.
- [71] D. Coffey, C. Lin, A. G. Erdman, and D. F. Keefe. Design by dragging: An interface for creative forward and inverse design with simulation ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2783–2791, Dec 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.147.
- [72] David Cohn, Rich Caruana, and Andrew McCallum. Semi-supervised clustering with user feedback. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, 4(1):17–32, 2003.
- [73] Danilo B Coimbra, Rafael M Martins, Tácito TAT Neves, Alexandru C Telea, and Fernando V Paulovich. Explaining three-dimensional dimensionality reduction plots. *Information Visualization*, 15(2):154–172, 2016.
- [74] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and*

- Computer Graphics*, 15(6):1009–1016, Nov 2009. ISSN 1077-2626. doi: 10.1109/TVCG.2009.122.
- [75] Paul G. Constantine. *Active Subspaces*. SIAM, Philadelphia, PA, 2015. doi: 10.1137/1.9781611973860.
- [76] Eric Corbett, Nathaniel Saul, and Meg Pirrung. Interactive machine learning heuristics. In *Proceedings of the Machine Learning from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.
- [77] Renato Cordeiro de Amorim and Peter Komisarczuk. *On Initializations for the Minkowski Weighted K-Means*, pages 45–55. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-34156-4. doi: 10.1007/978-3-642-34156-4_6. URL http://dx.doi.org/10.1007/978-3-642-34156-4_6.
- [78] Mark G Core, H Chad Lane, Michael Van Lent, Dave Gomboc, Steve Solomon, and Milton Rosenberg. Building explainable artificial intelligence systems. In *AAAI*, pages 1766–1773, 2006.
- [79] Jacqueline M Curiel and Gabriel A Radvansky. Mental organization of maps. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24(1):202, 1998.
- [80] S. Dash, A. Verma, C. North, and W. c. Feng. Portable parallel design of weighted multi-dimensional scaling for real-time data analysis. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 10–17, Dec 2017. doi: 10.1109/HPCC-SmartCity-DSS.2017.2.
- [81] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions*

- on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, April 1979. ISSN 0162-8828. doi: 10.1109/TPAMI.1979.4766909.
- [82] Renato Cordeiro de Amorim and Christian Hennig. Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences*, 324:126–145, 2015.
- [83] Çağatay Demiralp. Clustrophile: A tool for visual clustering analysis. In *Workshop on Interactive Data Exploration and Analytics*, IDEA '16, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-2138-9. doi: 10.1145/1235.
- [84] Marie Desjardins, James MacGlashan, and Julia Ferraioli. Interactive visual clustering. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 361–364. ACM, 2007.
- [85] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 269–274, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. doi: 10.1145/502512.502550. URL <http://doi.acm.org/10.1145/502512.502550>.
- [86] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1):143–175, 2001. ISSN 1573-0565. doi: 10.1023/A:1007612920971. URL <http://dx.doi.org/10.1023/A:1007612920971>.
- [87] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 89–98, New York, NY, USA, 2003. ACM. ISBN 1-58113-737-0. doi: 10.1145/956750.956764. URL <http://doi.acm.org/10.1145/956750.956764>.

- [88] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [89] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04*, pages 29–, New York, NY, USA, 2004. ACM. ISBN 1-58113-838-5. doi: 10.1145/1015330.1015408. URL <http://doi.acm.org/10.1145/1015330.1015408>.
- [90] Chris Ding and Tao Li. Adaptive dimension reduction using discriminant analysis and k-means clustering. In *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, pages 521–528, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273562. URL <http://doi.acm.org/10.1145/1273496.1273562>.
- [91] V. Dobrynin, D. Patterson, M. Galushka, and N. Rooney. Sophia: an interactive cluster-based retrieval system for the ohsumed collection. *IEEE Transactions on Information Technology in Biomedicine*, 9(2):256–265, June 2005. ISSN 1089-7771. doi: 10.1109/TITB.2005.847184.
- [92] David L Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. In *AMS Conference on Math Challenges of the 21st Century*, 2000.
- [93] E. P. dos Santos Amorim, E. V. Brazil, J. Daniels, P. Joia, L. G. Nonato, and M. C. Sousa. ilamp: Exploring high-dimensional spacing through backward multidimensional projection. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 53–62, Oct 2012. doi: 10.1109/VAST.2012.6400489.
- [94] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communi-*

- ation technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.
- [95] W. Dou, D. H. Jeong, F. Stukes, W. Ribarsky, H. R. Lipford, and R. Chang. Recovering reasoning processes from user interactions. *IEEE Computer Graphics and Applications*, 29(3):52–61, May 2009. ISSN 0272-1716. doi: 10.1109/MCG.2009.49.
- [96] Paul Dourish, John Lamping, and Tom Rodden. Building bridges: Customisation and mutual intelligibility in shared category management. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work, GROUP '99*, pages 11–20, New York, NY, USA, 1999. ACM. ISBN 1-58113-065-1. doi: 10.1145/320297.320299. URL <http://doi.acm.org/10.1145/320297.320299>.
- [97] Michelle Dowling, John Wenskovitch, Peter Hauck, Adam Binford, Nicholas Polys, and Chris North. A bidirectional pipeline for semantic interaction. In *Proceedings of the Workshop on Machine Learning from User Interaction for Visualization and Analytics, VIS 2018*, 2018.
- [98] Michelle Dowling, John Wenskovitch, J.T. Fry, Scotland Leman, Leanna House, and Chris North. Sirius: Dual, symmetric, interactive dimension reductions. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):172–182, Jan 2019. ISSN 1077-2626. doi: 10.1109/TVCG.2018.2865047.
- [99] Michelle Dowling, Nathan Wycoff, Brian Mayer, John Wenskovitch, Scotland Leman, Leanna House, Nicholas Polys, Chris North, and Peter Hauck. Interactive visual analytics for sensemaking with big text. *Big Data Research*, 16:49 – 58, 2019. ISSN 2214-5796. doi: <https://doi.org/10.1016/j.bdr.2019.04.003>. URL <http://www.sciencedirect.com/science/article/pii/S2214579618302995>.

- [100] Gregory Druck, Burr Settles, and Andrew McCallum. Active learning by labeling features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 81–90, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-59-6. URL <http://dl.acm.org/citation.cfm?id=1699510.1699522>.
- [101] Steven M. Drucker, Danyel Fisher, and Sumit Basu. Helping users sort faster with adaptive machine learning recommendations. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2011*, pages 187–203, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-23765-2.
- [102] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- [103] Avinava Dubey, Indrajit Bhattacharya, and Shantanu Godbole. A cluster-level semi-supervision model for interactive clustering. In José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 409–424, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15880-3.
- [104] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. doi: 10.1080/01969727308546046. URL <http://dx.doi.org/10.1080/01969727308546046>.
- [105] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.
- [106] T. Dwyer, Y. Koren, and K. Marriott. Isep-coala: An incremental procedure for sepa-

- ration constraint layout of graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):821–828, Sept 2006. ISSN 1077-2626. doi: 10.1109/TVCG.2006.156.
- [107] C. F. Eick, N. Zeidat, and Z. Zhao. Supervised clustering - algorithms and benefits. In *16th IEEE International Conference on Tools with Artificial Intelligence*, pages 774–776, Nov 2004. doi: 10.1109/ICTAI.2004.111.
- [108] M. El-Assady, R. Sevastjanova, F. Sperrle, D. Keim, and C. Collins. Progressive learning of topic modeling parameters: A visual analytics framework. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):382–391, Jan 2018. ISSN 1077-2626. doi: 10.1109/TVCG.2017.2745080.
- [109] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-level interaction with statistical models for visual analytics. In *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 121–130, Oct 2011. doi: 10.1109/VAST.2011.6102449.
- [110] A. Endert, P. Fiaux, and C. North. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2879–2888, Dec 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.260.
- [111] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *IEEE Computer Graphics and Applications*, 33(4):6–13, July 2013. ISSN 0272-1716. doi: 10.1109/MCG.2013.53.
- [112] A. Endert, R. Chang, C. North, and M. Zhou. Semantic interaction: Coupling cognition and computation through usable interactive analytics. *IEEE Computer Graphics and Applications*, 35(4):94–99, July 2015. ISSN 0272-1716. doi: 10.1109/MCG.2015.91.

- [113] Alex Endert. Semantic interaction for visual analytics: Toward coupling cognition and computation. *Computer Graphics and Applications, IEEE*, 34(4):8–15, July 2014. ISSN 0272-1716. doi: 10.1109/MCG.2014.73.
- [114] Alex Endert, Patrick Fiaux, and Chris North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 473–482, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4. doi: 10.1145/2207676.2207741. URL <http://doi.acm.org/10.1145/2207676.2207741>.
- [115] Alex Endert, Seth Fox, Dipayan Maiti, Scotland Leman, and Chris North. The semantics of clustering: Analysis of user-generated spatializations of text documents. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 555–562, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1287-5. doi: 10.1145/2254556.2254660. URL <http://doi.acm.org/10.1145/2254556.2254660>.
- [116] Alex Endert, M. Shahriar Hossain, Naren Ramakrishnan, Chris North, Patrick Fiaux, and Christopher Andrews. The human is the loop: new directions for visual analytics. *Journal of Intelligent Information Systems*, 43(3):411–435, 2014. ISSN 1573-7675. doi: 10.1007/s10844-014-0304-9. URL <http://dx.doi.org/10.1007/s10844-014-0304-9>.
- [117] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [118] Vladimir Estivill-Castro. Why so many clustering algorithms: A position paper. *SIGKDD Explor. Newsl.*, 4(1):65–75, June 2002. ISSN 1931-0145. doi: 10.1145/568574.568575. URL <http://doi.acm.org/10.1145/568574.568575>.
- [119] Brian Everitt. *Cluster Analysis*. Wiley, West Sussex, UK, 2011.

- [120] J. Faith. Targeted projection pursuit for interactive exploration of high- dimensional data sets. In *2007 11th International Conference Information Visualization (IV '07)*, pages 286–292, July 2007. doi: 10.1109/IV.2007.107.
- [121] Usama M Fayyad, Andreas Wierse, and Georges G Grinstein. *Information visualization in data mining and knowledge discovery*. Morgan Kaufmann, 2002.
- [122] Ronen Feldman and James Sanger. *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge university press, 2007.
- [123] Cristian Felix, Aritra Dasgupta, and Enrico Bertini. The exploratory labeling assistant: Mixed-initiative label curation with large document collections. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, UIST '18, pages 153–164, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5948-1. doi: 10.1145/3242587.3242596. URL <http://doi.acm.org/10.1145/3242587.3242596>.
- [124] I K Fodor. *A Survey of Dimension Reduction Techniques*. Los Alamos National Lab, May 2002. doi: 10.2172/15002155. URL <http://www.osti.gov/scitech/servlets/purl/15002155>.
- [125] Edward B Fowlkes and Colin L Mallows. A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, 78(383):553–569, 1983.
- [126] S. L. France and J. D. Carroll. Two-way multidimensional scaling: A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(5):644–661, Sept 2011. ISSN 1094-6977. doi: 10.1109/TSMCC.2010.2078502.
- [127] J. H. Friedman and J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23(9):881–890, Sept 1974. ISSN 0018-9340. doi: 10.1109/T-C.1974.224051.

- [128] Jerome H Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.
- [129] E. R. Gansner, Y. Hu, and S. Kobourov. Gmap: Visualizing graphs and clusters as maps. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 201–208, March 2010. doi: 10.1109/PACIFICVIS.2010.5429590.
- [130] M. Garey, D. Johnson, and H. Witsenhausen. The complexity of the generalized lloyd - max problem (corresp.). *IEEE Transactions on Information Theory*, 28(2):255–256, March 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056488.
- [131] A. Geva and I. Gath. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 11(07):773–780, jul 1989. ISSN 0162-8828. doi: 10.1109/34.192473.
- [132] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [133] A.K. Gorban, Balázs Kégl, Donald Wunsch, and Andrei Zinovyev. *Principal Manifolds for Data Visualisation and Dimension Reduction*. Springer, 01 2008. ISBN 978-3-540-73750-6.
- [134] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27:857–871, 1971.
- [135] Ronald L Graham and F Frances Yao. Finding the convex hull of a simple polygon. *Journal of Algorithms*, 4(4):324–331, 1983.
- [136] David Gunning. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA)*, nd Web, 2017.

- [137] P. Guo, H. Xiao, Z. Wang, and X. Yuan. Interactive local clustering operations for high dimensional data in parallel coordinates. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 97–104, March 2010. doi: 10.1109/PACIFICVIS.2010.5429608.
- [138] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, March 2003. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=944919.944968>.
- [139] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1):389–422, Jan 2002. ISSN 1573-0565. doi: 10.1023/A:1012487302797. URL <https://doi.org/10.1023/A:1012487302797>.
- [140] M. Halkidi and M. Vazirgiannis. Clustering validity assessment: finding the optimal partitioning of a data set. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 187–194, Nov 2001. doi: 10.1109/ICDM.2001.989517.
- [141] Peter Hamilton and Daniel J. Wigdor. Conductor: Enabling and understanding cross-device interaction. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems, CHI '14*, pages 2773–2782, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2473-1. doi: 10.1145/2556288.2557170. URL <http://doi.acm.org/10.1145/2556288.2557170>.
- [142] Harry H Harman. Modern factor analysis. *University of Chicago Press*, 1960.
- [143] J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972. doi: 10.1080/01621459.1972.10481214.
- [144] J. A. Hartigan. *Cluster Analysis*. John Wiley & Sons, New York, 1975.

- [145] Rex Hartson and Pardha S Pyla. *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier, 2012.
- [146] Reid Hastie and Purohit A Kumar. Person memory: Personality traits as organizing principles in memory for behaviors. *Journal of Personality and Social Psychology*, 37(1):25, 1979.
- [147] Christopher G. Healey, Kellogg S. Booth, and James T. Enns. High-speed visual estimation using preattentive processing. *ACM Trans. Comput.-Hum. Interact.*, 3(2):107–135, June 1996. ISSN 1073-0516. doi: 10.1145/230562.230563. URL <http://doi.acm.org/10.1145/230562.230563>.
- [148] J. Heer and D. Boyd. Vizster: visualizing online social networks. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pages 32–39, Oct 2005. doi: 10.1109/INFVIS.2005.1532126.
- [149] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, Nov 2008. ISSN 1077-2626. doi: 10.1109/TVCG.2008.137.
- [150] Christian Heine and Gerik Scheuermann. Manual clustering refinement using interaction with blobs. In *Proceedings of the 9th Joint Eurographics / IEEE VGTC Conference on Visualization, EUROVIS'07*, pages 59–66, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. ISBN 978-3-905673-45-6. doi: 10.2312/VisSym/EuroVis07/059-066. URL <http://dx.doi.org/10.2312/VisSym/EuroVis07/059-066>.
- [151] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

- [152] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 289–296, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-614-9. URL <http://dl.acm.org/citation.cfm?id=2073796.2073829>.
- [153] Sidney P Holman. Entropy and insight: Exploring how information theory can be used to quantify sensemaking in visual analytics. Master's thesis, Virginia Tech, 2018.
- [154] Enamul Hoque and Giuseppe Carenini. Convisit: Interactive topic modeling for exploring asynchronous online conversations. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 169–180, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3306-1. doi: 10.1145/2678025.2701370. URL <http://doi.acm.org/10.1145/2678025.2701370>.
- [155] M. S. Hossain, P. K. R. Ojili, C. Grimm, R. Mller, L. T. Watson, and N. Ramakrishnan. Scatter/gather clustering: Flexibly incorporating user feedback to steer clustering results. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2829–2838, Dec 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.258.
- [156] Michael E. Houle, Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Can shared-neighbor distances defeat the curse of dimensionality? In Michael Gertz and Bertram Ludäscher, editors, *Scientific and Statistical Database Management*, pages 482–500, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-13818-8.
- [157] Leanna House, Scotland Leman, and Chao Han. Bayesian visual analytics: Bava. *Statistical Analysis and Data Mining*, 8(1):1–13, 2015. ISSN 1932-1872. doi: 10.1002/sam.11253. URL <http://dx.doi.org/10.1002/sam.11253>.

- [158] X. Hu, L. Bradel, D. Maiti, L. House, C. North, and S. Leman. Semantics of directly manipulating spatializations. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2052–2059, Dec 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.188.
- [159] Yeming Hu, Evangelos E. Milios, James Blustein, and Shali Liu. Personalized document clustering with dual supervision. In *Proceedings of the 2012 ACM Symposium on Document Engineering*, DocEng '12, pages 161–170, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1116-8. doi: 10.1145/2361354.2361393. URL <http://doi.acm.org/10.1145/2361354.2361393>.
- [160] Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. Interactive topic modeling. *Machine Learning*, 95(3):423–469, Jun 2014. ISSN 1573-0565. doi: 10.1007/s10994-013-5413-0. URL <https://doi.org/10.1007/s10994-013-5413-0>.
- [161] A. Huang, D. Milne, E. Frank, and I. H. Witten. Clustering documents with active learning using wikipedia. In *2008 Eighth IEEE International Conference on Data Mining*, pages 839–844, Dec 2008. doi: 10.1109/ICDM.2008.80.
- [162] J. Z. Huang, M. K. Ng, Hongqiang Rong, and Zichen Li. Automated variable weighting in k-means type clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):657–668, May 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.95.
- [163] Mary Inaba, Naoki Katoh, and Hiroshi Imai. Applications of weighted voronoi diagrams and randomization to variance-based k-clustering: (extended abstract). In *Proceedings of the Tenth Annual Symposium on Computational Geometry*, SCG '94, pages 332–339, New York, NY, USA, 1994. ACM. ISBN 0-89791-648-4. doi: 10.1145/177424.178042. URL <http://doi.acm.org/10.1145/177424.178042>.
- [164] S. Ingram, T. Munzner, and M. Olano. Glimmer: Multilevel mds on the gpu. *IEEE*

- Transactions on Visualization and Computer Graphics*, 15(2):249–261, March 2009. ISSN 1077-2626. doi: 10.1109/TVCG.2008.85.
- [165] Petra Isenberg, Pierre Dragicevic, and Yvonne Jansen. Classic datasets – cereals. <https://perso.telecom-paristech.fr/eagan/class/igr204/datasets>, 2019.
- [166] Tomoharu Iwata, Takeshi Yamada, and Naonori Ueda. Probabilistic latent semantic visualization: Topic model for visualizing documents. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 363–371, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401937. URL <http://doi.acm.org/10.1145/1401890.1401937>.
- [167] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.
- [168] R. Jianu, A. Rusu, Y. Hu, and D. Taggart. How to display group information on node-link diagrams: An evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 20(11):1530–1541, Nov 2014. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2315995.
- [169] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato. Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2563–2571, Dec 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2011.220.
- [170] P. Joia, F. Petronetto, and L. G. Nonato. Uncovering representative groups in multi-dimensional projections. In *Proceedings of the 2015 Eurographics Conference on Visualization*, EuroVis '15, pages 281–290, Aire-la-Ville, Switzerland, Switzerland, 2015. Eurographics Association. doi: 10.1111/cgf.12640. URL <http://dx.doi.org/10.1111/cgf.12640>.

- [171] Ian Jolliffe. *Principal Component Analysis*, pages 1094–1096. John Wiley & Sons, Ltd, 2014. ISBN 9781118445112. doi: 10.1002/9781118445112.stat06472. URL <http://dx.doi.org/10.1002/9781118445112.stat06472>.
- [172] Thouis R Jones, Anne E Carpenter, Michael R Lamprecht, Jason Moffat, Serena J Silver, Jennifer K Grenier, Adam B Castoreno, Ulrike S Eggert, David E Root, Polina Golland, et al. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning. *Proceedings of the National Academy of Sciences*, 106(6):1826–1831, 2009.
- [173] Karin Kailing, Hans-Peter Kriegel, and Peer Krger. *Density-Connected Subspace Clustering for High-Dimensional Data*, pages 246–256. Society for Industrial and Applied Mathematics, 2004. doi: 10.1137/1.9781611972740.23. URL <https://epubs.siam.org/doi/abs/10.1137/1.9781611972740.23>.
- [174] E. Kandogan. Star coordinate: A multi-dimensional visualization technique with uniform treatment of dimensions. In *Proceedings of the IEEE Information Visualization Symposium*, volume 650, page 22, 2000.
- [175] E. Kandogan. Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 73–82, Oct 2012. doi: 10.1109/VAST.2012.6400487.
- [176] Juha Karhunen, Petteri Pajunen, and Erkki Oja. The nonlinear {PCA} criterion in blind source separation: Relations with other approaches. *Neurocomputing*, 22(13): 5–20, 1998. ISSN 0925-2312. doi: [http://dx.doi.org/10.1016/S0925-2312\(98\)00046-0](http://dx.doi.org/10.1016/S0925-2312(98)00046-0). URL <http://www.sciencedirect.com/science/article/pii/S0925231298000460>.
- [177] Andrej Karpathy. t-sne csv web demo. <https://cs.stanford.edu/people/karpathy/tsnejs/csvdemo.html>, 2016.

- [178] Leonard Kaufman and Peter J. Rousseeuw. Clustering by means of medoids. *Statistical Data Analysis Based on the L1-Norm and Related Methods*, pages 405–416, 1987.
- [179] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. *Visual Analytics: Definition, Process, and Challenges*, pages 154–175. Springer, Heidelberg, 2008. ISBN 978-3-540-70956-5. doi: 10.1007/978-3-540-70956-5_7. URL http://dx.doi.org/10.1007/978-3-540-70956-5_7.
- [180] H. Kim, J. Choo, H. Park, and A. Endert. Interaxis: Steering scatterplot axes via observation-level interaction. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):131–140, Jan 2016. ISSN 1077-2626. doi: 10.1109/TVCG.2015.2467615.
- [181] M. Kim, K. Kang, D. Park, J. Choo, and N. Elmqvist. Topiclens: Efficient multi-level visual topic exploration of large-scale document collections. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):151–160, Jan 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2598445.
- [182] Gary Klein, Brian Moon, and Robert R Hoffman. Making sense of sensemaking 2: A macrocognitive model. *IEEE Intelligent systems*, 21(5):88–92, 2006.
- [183] Tijmen Klein, Florimond Guéniat, Luc Pastur, Frédéric Vernier, and Tobias Isenberg. A design study of direct-touch interaction for exploratory 3d scientific visualization. In *Computer Graphics Forum*, volume 31, pages 1225–1234. Wiley Online Library, 2012.
- [184] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, Sep 1990. ISSN 0018-9219. doi: 10.1109/5.58325.
- [185] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Bio-*

- logical Cybernetics*, 43(1):59–69, Jan 1982. ISSN 1432-0770. doi: 10.1007/BF00337288.
URL <https://doi.org/10.1007/BF00337288>.
- [186] Yuki Koyama, Daisuke Sakamoto, and Takeo Igarashi. Selph: Progressive learning and support of manual photo color enhancement. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2520–2532, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858111. URL <http://doi.acm.org/10.1145/2858036.2858111>.
- [187] Josua Krause, Adam Perer, and Kenney Ng. Interacting with predictions: Visual inspection of black-box machine learning models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5686–5697, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3362-7. doi: 10.1145/2858036.2858529. URL <http://doi.acm.org/10.1145/2858036.2858529>.
- [188] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Trans. Knowl. Discov. Data*, 3(1):1:1–1:58, March 2009. ISSN 1556-4681. doi: 10.1145/1497577.1497578. URL <http://doi.acm.org/10.1145/1497577.1497578>.
- [189] Hans-Peter Kriegel, Peer Krger, Jrg Sander, and Arthur Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011. doi: 10.1002/widm.30. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.30>.
- [190] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Subspace clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(4):351–364, 2012.
- [191] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

- [192] Joseph B Kruskal. Toward a practical method which helps uncover the structure of a set of multivariate observations by finding the linear transformation which optimizes a new index of condensation. In *Statistical Computation*, pages 427–440. Elsevier, 1969.
- [193] Joseph B Kruskal and Myron Wish. Multidimensional scaling. *Quantitative Applications in the social Sciences Series, Newbury Park: Sage Publications*, 11, 1978.
- [194] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):221–230, Jan 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2598446.
- [195] Christoph H Lampert, Hannes Nickisch, Stefan Harmeling, and Jens Weidmann. Animals with attributes: A dataset for attribute based classification, 2009.
- [196] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. ivis-clustering: An interactive visual document clustering via topic modeling. *Computer Graphics Forum*, 31(3pt3):1155–1164, 2012. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2012.03108.x. URL <http://dx.doi.org/10.1111/j.1467-8659.2012.03108.x>.
- [197] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.
- [198] Tak Yeon Lee, Alison Smith, Kevin Seppi, Niklas Elmqvist, Jordan Boyd-Graber, and Leah Findlater. The human touch: How non-expert users perceive, interpret, and fix topic models. *International Journal of Human-Computer Studies*, 105:28–42, 2017.
- [199] Scotland C. Leman, Leanna House, Dipayan Maiti, Alex Endert, and Chris North. Visual to parametric interaction (v2pi). *PLoS ONE*, 8(3):1–12, 03 2013. doi: 10.1371/journal.pone.0050474. URL <http://dx.doi.org/10.1371%2Fjournal.pone.0050474>.

- [200] Michael Levandowsky and David Winter. Distance between sets. *Nature*, 234(5323): 34–35, 1971.
- [201] Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991. doi: 10.2307/2290563.
- [202] Ker-Chau Li. On principal Hessian directions for data visualization and dimension reduction: Another application of Stein’s lemma. *Journal of the American Statistical Association*, 87(420):1025–1039, 1992. doi: 10.2307/2290640.
- [203] Tianyi Li, Gregorio Convertino, Wenbo Wang, Haley Most, Tristan Zajonc, and Yi-Hsun Tsai. Hypertuner: Visual analytics for hyperparameter tuning by professionals. In *Proceedings of the Machine Learning from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.
- [204] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84 – 96, 2014. ISSN 0925-2312. doi: <https://doi.org/10.1016/j.neucom.2013.09.055>. URL <http://www.sciencedirect.com/science/article/pii/S0925231214003658>.
- [205] S. Liu, B. Wang, P.-T. Bremer, and V. Pascucci. Distortion-guided structure-driven interactive exploration of high-dimensional data. *Computer Graphics Forum*, 33(3): 101–110, 2014. ISSN 1467-8659. doi: 10.1111/cgf.12366. URL <http://dx.doi.org/10.1111/cgf.12366>.
- [206] S. Liu, X. Wang, J. Chen, J. Zhu, and B. Guo. Topicpanorama: A full picture of relevant topics. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 183–192, Oct 2014. doi: 10.1109/VAST.2014.7042494.

- [207] S. Liu, D. Maljovec, B. Wang, P. T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, March 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2640960.
- [208] Shixia Liu, Michelle X. Zhou, Shimei Pan, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Interactive, topic-based visual text summarization and analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09*, pages 543–552, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-512-3. doi: 10.1145/1645953.1646023. URL <http://doi.acm.org/10.1145/1645953.1646023>.
- [209] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, Mar 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489.
- [210] Haiping Lu, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540 – 1551, 2011. ISSN 0031-3203. doi: <https://doi.org/10.1016/j.patcog.2011.01.004>. URL <http://www.sciencedirect.com/science/article/pii/S0031320311000136>.
- [211] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9:2579–2605, September 2008.
- [212] J. MacInnes, S. Santosa, and W. Wright. Visual classification: Expert knowledge guides machine learning. *IEEE Computer Graphics and Applications*, 30(1):8–14, Jan 2010. ISSN 0272-1716. doi: 10.1109/MCG.2010.18.
- [213] Thomas W. Malone. How do people organize their desks?: Implications for the design of office information systems. *ACM Trans. Inf. Syst.*, 1(1):99–112, January 1983. ISSN 1046-8188. doi: 10.1145/357423.357430. URL <http://doi.acm.org/10.1145/357423.357430>.

- [214] Gladys MH Mamani, Francisco M Fatore, Luis Gustavo Nonato, and Fernando Vieira Paulovich. User-driven feature space transformation. *Computer Graphics Forum*, 32(3pt3):291–299, 2013. ISSN 1467-8659. doi: 10.1111/cgf.12116. URL <http://dx.doi.org/10.1111/cgf.12116>.
- [215] Richard Mander, Gitta Salomon, and Yin Yin Wong. A “pile” metaphor for supporting casual organization of information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’92*, pages 627–634, New York, NY, USA, 1992. ACM. ISBN 0-89791-513-5. doi: 10.1145/142750.143055. URL <http://doi.acm.org/10.1145/142750.143055>.
- [216] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. *Natural Language Engineering*, 16(1):100–103, 2010.
- [217] G. E. Marai, A. G. Forbes, and A. Johnson. Interdisciplinary immersive analytics at the electronic visualization laboratory: Lessons learned and upcoming challenges. In *2016 Workshop on Immersive Analytics (IA)*, pages 54–59, March 2016. doi: 10.1109/IMMERSIVE.2016.7932384.
- [218] A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1526–1538, Sept 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2013.65.
- [219] Geoffrey McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004.
- [220] Geoffrey J McLachlan and Kaye E Basford. *Mixture models: Inference and applications to clustering*, volume 84. M. Dekker New York, 1988.

- [221] Tauno Metsalu and Jaak Vilo. Clustvis: a web tool for visualizing clustering of multivariate data using principal component analysis and heatmap. *Nucleic acids research*, 43(W1):W566–W570, 2015.
- [222] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 2018.
- [223] Boris Mirkin. *Mathematical classification and clustering*. Kluwer Academic Publishers, 1996.
- [224] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyama. Layout adjustment and the mental map. *Journal of Visual Languages & Computing*, 6(2):183 – 210, 1995. ISSN 1045-926X. doi: <http://dx.doi.org/10.1006/jvlc.1995.1010>. URL <http://www.sciencedirect.com/science/article/pii/S1045926X85710105>.
- [225] Vladimir Molchanov and Lars Linsen. Interactive Design of Multidimensional Data Projection Layout. In N. Elmqvist, M. Hlawitschka, and J. Kennedy, editors, *EuroVis - Short Papers*. The Eurographics Association, 2014. ISBN 978-3-905674-69-9. doi: 10.2312/eurovisshort.20141152.
- [226] T. K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, Nov 1996. ISSN 1053-5888. doi: 10.1109/79.543975.
- [227] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.
- [228] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre. Clustersculptor: A visual analytics tool for high-dimensional data. In *2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 75–82, Oct 2007. doi: 10.1109/VAST.2007.4388999.
- [229] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14, pages 849–856, 2001.

- [230] Jakob Nielsen. Iterative user-interface design. *Computer*, 26(11):32–41, 1993.
- [231] Janni Nielsen, Torkil Clemmensen, and Carsten Yssing. Getting access to what goes on in people’s heads?: Reflections on the think-aloud technique. In *Proceedings of the Second Nordic Conference on Human-computer Interaction, NordiCHI ’02*, pages 101–110, New York, NY, USA, 2002. ACM. ISBN 1-58113-616-1. doi: 10.1145/572020.572033. URL <http://doi.acm.org/10.1145/572020.572033>.
- [232] Donglin Niu, Jennifer G Dy, and Michael I Jordan. Dimensionality reduction for spectral clustering. In *Proceedings of the 14th International Conference Artificial Intelligence and Statistics, AISTATS ’11*, pages 552–560, New York, NY, USA, 2011. ACM.
- [233] Chris North. Toward measuring visualization insight. *IEEE computer graphics and applications*, 26(3):6–9, 2006.
- [234] D. Orban, D. F. Keefe, A. Biswas, J. Ahrens, and D. Rogers. Drag and track: A direct manipulation interface for contextualizing data instances within a continuous parameter space. *IEEE Transactions on Visualization and Computer Graphics*, 25(1): 256–266, Jan 2019. ISSN 1077-2626. doi: 10.1109/TVCG.2018.2865051.
- [235] Alvitta Ottley, Huahai Yang, and Remco Chang. Personality as a predictor of user strategy: How locus of control affects search strategies on tree visualizations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI ’15*, pages 3251–3254, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3145-6. doi: 10.1145/2702123.2702590. URL <http://doi.acm.org/10.1145/2702123.2702590>.
- [236] Malay K Pakhira, Sanghamitra Bandyopadhyay, and Ujjwal Maulik. Validity index for crisp and fuzzy clusters. *Pattern recognition*, 37(3):487–501, 2004.

- [237] Prateek Panwar, Adam Bradley, and Christopher Collins. Providing contextual assistance in response to frustration in visual analytics tasks. In *Proceedings of the Machine Learning from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.
- [238] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, pages 159–168, New York, NY, USA, 1998. ACM. ISBN 0-89791-996-3. doi: 10.1145/275487.275505. URL <http://doi.acm.org/10.1145/275487.275505>.
- [239] F.V. Paulovich, D.M. Eler, J. Poco, C.P. Botha, R. Minghim, and L.G. Nonato. Piecewise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, 30(3):1091–1100, 2011. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2011.01958.x. URL <http://dx.doi.org/10.1111/j.1467-8659.2011.01958.x>.
- [240] Karl Pearson. Principal components analysis. *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, 6(2):566, 1901.
- [241] Ren Peeters. The maximum edge biclique problem is np-complete. *Discrete Applied Mathematics*, 131(3):651 – 654, 2003. ISSN 0166-218X. doi: [https://doi.org/10.1016/S0166-218X\(03\)00333-0](https://doi.org/10.1016/S0166-218X(03)00333-0). URL <http://www.sciencedirect.com/science/article/pii/S0166218X03003330>.
- [242] Dan Pelleg, Andrew W Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, volume 1, pages 727–734, 2000.
- [243] Fabian C. Peña and John Guerra-Gomez. Opening the black-box: Towards more interactive and interpretable machine learning. In *Proceedings of the Machine Learning*

- from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.
- [244] Darius Pfitzner, Richard Leibbrandt, and David Powers. Characterization and evaluation of similarity measures for pairs of clusterings. *Knowledge and Information Systems*, 19(3):361, 2009.
- [245] T. M. Phuong, Z. Lin, and R. B. Altman. Choosing snps using feature selection. In *2005 IEEE Computational Systems Bioinformatics Conference (CSB'05)*, pages 301–309, Aug 2005. doi: 10.1109/CSB.2005.22.
- [246] Peter Pirolli. A theory of information scent. *Human-computer interaction*, 1:213–217, 2003.
- [247] Peter Pirolli and Stuart Card. Information foraging. *Psychological review*, 106(4):643, 1999.
- [248] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. *Proceedings of International Conference on Intelligence Analysis*, 5:2–4, 2005.
- [249] E. D. Ragan, A. Endert, J. Sanyal, and J. Chen. Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, Jan 2016. ISSN 1077-2626. doi: 10.1109/TVCG.2015.2467551.
- [250] Hema Raghavan and James Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information*

- Retrieval*, SIGIR '07, pages 79–86, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-597-7. doi: 10.1145/1277741.1277758. URL <http://doi.acm.org/10.1145/1277741.1277758>.
- [251] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.
- [252] B. Rieck and H. Leitte. Persistent homology for the evaluation of dimensionality reduction schemes. *Computer Graphics Forum*, 34(3):431–440, 2015. ISSN 1467-8659. doi: 10.1111/cgf.12655. URL <http://dx.doi.org/10.1111/cgf.12655>.
- [253] Duda Ro and Hart Pe. *Pattern classification and scene analysis*. John Wiley & Sons, New York, USA, 1973.
- [254] George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data mountain: Using spatial memory for document management. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 153–162, New York, NY, USA, 1998. ACM. ISBN 1-58113-034-1. doi: 10.1145/288392.288596. URL <http://doi.acm.org/10.1145/288392.288596>.
- [255] Yvonne Rogers and Judi Ellis. Distributed cognition: an alternative framework for analysing and explaining collaborative working. *Journal of Information Technology*, 9(2):119–128, Jun 1994. ISSN 1466-4437. doi: 10.1057/jit.1994.12. URL <https://doi.org/10.1057/jit.1994.12>.
- [256] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.
- [257] Hugo Romat, Nathalie Henry Riche, Ken Hinckley, Bongshin Lee, Caroline Appert,

- Emmanuel Pietriga, and Christopher Collins. Activeink: (th)inking with data. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 42:1–42:13, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-5970-2. doi: 10.1145/3290605.3300272. URL <http://doi.acm.org/10.1145/3290605.3300272>.
- [258] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [259] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. ISSN 0036-8075. doi: 10.1126/science.290.5500.2323. URL <https://science.sciencemag.org/content/290/5500/2323>.
- [260] Adam Rule, Aurelien Tabard, and James D. Hollan. Data from: Exploration and explanation in computational notebooks. UC San Diego Library Digital Collections, 2018.
- [261] David E Rumelhart and David Zipser. Feature discovery by competitive learning. *Cognitive science*, 9(1):75–112, 1985.
- [262] Tuukka Ruotsalo, Jaakko Peltonen, Manuel Eugster, Dorota Glowacka, Ksenia Konyushkova, Kumaripaba Athukorala, Ilkka Kosunen, Aki Reijonen, Petri Myllymäki, Giulio Jacucci, and Samuel Kaski. Directing exploratory search with interactive intent modeling. In *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management*, CIKM '13, pages 1759–1764, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2263-8. doi: 10.1145/2505515.2505644. URL <http://doi.acm.org/10.1145/2505515.2505644>.
- [263] Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. The cost structure of sensemaking. In *Proceedings of the INTERACT '93 and CHI '93 Confer-*

- ence on Human Factors in Computing Systems*, CHI '93, pages 269–276, New York, NY, USA, 1993. ACM. ISBN 0-89791-575-5. doi: 10.1145/169059.169209. URL <http://doi.acm.org/10.1145/169059.169209>.
- [264] D. Sacha, L. Zhang, M. Sedlmair, J. A. Lee, J. Peltonen, D. Weiskopf, S. C. North, and D. A. Keim. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):241–250, Jan 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2598495.
- [265] B. Saket, P. Simonetto, S. Kobourov, and K. Brner. Node, node-link, and node-link-group diagrams: An evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2231–2240, Dec 2014. ISSN 1077-2626. doi: 10.1109/TVCG.2014.2346422.
- [266] Alexander M. Samarov. Exploring regression structure using nonparametric functional estimation. *Journal of the American Statistical Association*, 88(423):836–847, 1993. doi: 10.2307/2290772.
- [267] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*, 2017.
- [268] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead. Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, Nov 2010. ISSN 1077-2626. doi: 10.1109/TVCG.2010.181.
- [269] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):443–456, July 2005. ISSN 1077-2626. doi: 10.1109/TVCG.2005.53.

- [270] Susan S Schiffman, Forrest W Young, and M Lance Reynolds. *Introduction to multi-dimensional scaling: Theory, methods, and applications*. Taylor Francis, 1981.
- [271] Steven Schwartz. Individual differences in cognition: Some relationships between personality and memory. *Journal of Research in Personality*, 9(3):217–225, 1975.
- [272] Jessica Zeitz Self, Xinran Hu, Leanna House, Scotland Leman, and Chris North. Designing usable interactive visual analytics tools for dimension reduction. In *CHI 2016 Workshop on Human-Centered Machine Learning (HCML)*, page 7, May 2016.
- [273] Jessica Zeitz Self, Radha Krishnan Vinayagam, J. T. Fry, and Chris North. Bridging the gap between user intention and model parameters for human-in-the-loop data analytics. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics, HILDA '16*, pages 3:1–3:6, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4207-0. doi: 10.1145/2939502.2939505. URL <http://doi.acm.org/10.1145/2939502.2939505>.
- [274] Jessica Zeitz Self, Michelle Dowling, John Wenskovitch, Ian Crandell, Ming Wang, Leanna House, Scotland Leman, and Chris North. Observation-level and parametric interaction for high-dimensional data analysis. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 8(2):15:1–15:36, June 2018. ISSN 2160-6455. doi: 10.1145/3158230. URL <http://doi.acm.org/10.1145/3158230>.
- [275] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.
- [276] John Sharko, Georges Grinstein, and Kenneth A Marx. Vectorized radviz and its application to multiple cluster datasets. *IEEE transactions on Visualization and Computer Graphics*, 14(6), 2008.
- [277] Frank M Shipman III and Raymond McCall. Supporting knowledge-base evolution

- with incremental formalization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 285–291, 1994.
- [278] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*, pages 364–371. Elsevier, 2003.
- [279] Robert Simmon. Use of color in data visualization. https://earthobservatory.nasa.gov/resources/blogs/intro_to_color_for_visualization.pdf, 2013.
- [280] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.
- [281] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg. Embedding Projector: Interactive Visualization and Interpretation of Embeddings. *arXiv e-prints*, November 2016.
- [282] Olga Sourina and Dongquan Liu. Visual interactive clustering and querying of spatio-temporal data. In Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganá, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan, editors, *Computational Science and Its Applications – ICCSA 2005*, pages 968–977, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-32309-9.
- [283] Fabian Sperrle, Jürgen Bernard, Michael Sedlmair, Daniel Keim, and Mennatallah El-Assady. Speculative execution for guided visual analytics. In *Proceedings of the Machine Learning from User Interaction for Visualization and Analytics Workshop at IEEE VIS 2018*, Oct 2018.
- [284] J. Stahnke, M. Drk, B. Mller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE*

- Transactions on Visualization and Computer Graphics*, 22(1):629–638, Jan 2016. ISSN 1077-2626. doi: 10.1109/TVCG.2015.2467717.
- [285] John Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information visualization*, 7(2):118–132, 2008.
- [286] Wolfgang Stuerzlinger, Tim Dwyer, Steven Drucker, Carsten Görg, Chris North, and Gerek Scheuermann. Immersive human-centered computational analytics. In *Immersive Analytics*, pages 139–163. Springer, 2018.
- [287] Simone Stumpf, Vidya Rajaram, Lida Li, Margaret Burnett, Thomas Dietterich, Erin Sullivan, Russell Drummond, and Jonathan Herlocker. Toward harnessing user feedback for machine learning. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 82–91. ACM, 2007.
- [288] Catherine A Sugar and Gareth M James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763, 2003.
- [289] Gabor J Szekely and Maria L Rizzo. Hierarchical clustering via joint between-within distances: Extending ward’s minimum variance method. *Journal of classification*, 22(2):151–183, 2005.
- [290] Paolo Tamagnini, Josua Krause, Aritra Dasgupta, and Enrico Bertini. Interpreting black-box classifiers using instance-level visual explanations. In *Proceedings of the 2Nd Workshop on Human-In-the-Loop Data Analytics, HILDA’17*, pages 6:1–6:6, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5029-7. doi: 10.1145/3077257.3077260. URL <http://doi.acm.org/10.1145/3077257.3077260>.
- [291] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. Data mining cluster analysis:

- basic concepts and algorithms. In *Introduction to data mining*, chapter 8. Pearson Education India, 2013.
- [292] Yee Whye Teh. Dirichlet process. In *Encyclopedia of machine learning*, pages 280–287. Springer, 2011.
- [293] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392, 2005.
- [294] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. ISSN 0036-8075. doi: 10.1126/science.290.5500.2319. URL <http://science.sciencemag.org/content/290/5500/2319>.
- [295] Bruce Thompson. *Exploratory and confirmatory factor analysis: Understanding concepts and applications*. American Psychological Association, 2004.
- [296] Robert L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, Dec 1953. ISSN 1860-0980. doi: 10.1007/BF02289263. URL <https://doi.org/10.1007/BF02289263>.
- [297] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.
- [298] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.
- [299] Michael E Tipping and Christopher M Bishop. Probabilistic principal component

- analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.
- [300] Warren S Torgerson. *Theory and methods of scaling*. Wiley, Oxford, England, 1958.
- [301] G. V. Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):306–307, July 1979. ISSN 0162-8828. doi: 10.1109/TPAMI.1979.4766926.
- [302] John W Tukey. The future of data analysis. *The annals of mathematical statistics*, 33(1):1–67, 1962.
- [303] C. Turkey, P. Filzmoser, and H. Hauser. Brushing dimensions - a dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2591–2599, Dec 2011. ISSN 1077-2626. doi: 10.1109/TVCG.2011.178.
- [304] United States Census Bureau. Census Regions and Divisions of the United States, 2016. URL http://www2.census.gov/geo/pdfs/maps-data/maps/reference/us_regdiv.pdf.
- [305] United States Census Bureau. State Data Center Program, 2019. URL <https://www.census.gov/about/partners/sdc.html>.
- [306] Eliane R. A. Valiati, Marcelo S. Pimenta, and Carla M. D. S. Freitas. A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–6, New York, NY, USA, 2006. ACM. ISBN 1-59593-562-2. doi: 10.1145/1168149.1168169. URL <http://doi.acm.org/10.1145/1168149.1168169>.

- [307] A. van Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola, and R. M. Simpson. Immersive vr for scientific visualization: a progress report. *IEEE Computer Graphics and Applications*, 20(6):26–52, Nov 2000. ISSN 0272-1716. doi: 10.1109/38.888006.
- [308] Andrea Vattani. The hardness of k-means clustering in the plane. *UCSD*, 2009.
- [309] Jarkko Venna and Samuel Kaski. Local multidimensional scaling. *Neural Networks*, 19(6-7):889–899, 2006.
- [310] Tatiana von Landesberger, Sebastian Fiebig, Sebastian Bremm, Arjan Kuijper, and Dieter W. Fellner. *Interaction Taxonomy for Tracking of User Actions in Visual Analytics Applications*, pages 653–670. Springer New York, New York, NY, 2014. ISBN 978-1-4614-7485-2. doi: 10.1007/978-1-4614-7485-2_26. URL https://doi.org/10.1007/978-1-4614-7485-2_26.
- [311] J. A. Wagner Filho, M. F. Rey, C. M. D. S. Freitas, and L. Nedel. Immersive visualization of abstract information: An evaluation on dimensionally-reduced data scatterplots. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 483–490, March 2018. doi: 10.1109/VR.2018.8447558.
- [312] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, volume 1, pages 577–584, 2001.
- [313] E. Wall, S. Das, R. Chawla, B. Kalidindi, E. T. Brown, and A. Endert. Podium: Ranking data using mixed-initiative visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):288–297, Jan 2018. ISSN 1077-2626. doi: 10.1109/TVCG.2017.2745078.
- [314] Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd*

- International Conference on Machine Learning*, ICML '06, pages 977–984, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2. doi: 10.1145/1143844.1143967. URL <http://doi.acm.org/10.1145/1143844.1143967>.
- [315] Colin Ware, David Rogers, Mark Petersen, James Ahrens, and Erol Aygar. Optimizing for visual cognition in high performance scientific computing. *Electronic Imaging*, 2016 (16):1–9, 2016.
- [316] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016. doi: 10.23915/distill.00002. URL <http://distill.pub/2016/misread-tsne>.
- [317] John Wenskovitch and Chris North. Observation-level interaction with clustering and dimension reduction algorithms. In *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, HILDA'17, pages 14:1–14:6, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5029-7. doi: 10.1145/3077257.3077259. URL <http://doi.acm.org/10.1145/3077257.3077259>.
- [318] John Wenskovitch and Chris North. Pollux: Interactive cluster-first projections of high-dimensional data. In *2019 Symposium on Visualization in Data Science*, VIS 2019, 2019.
- [319] John Wenskovitch, Lauren Bradel, Michelle Dowling, Leanna House, and Chris North. The effect of semantic interaction on foraging in text analysis. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2018.
- [320] John Wenskovitch, Ian Crandell, Naren Ramakrishnan, Leanna House, Scotland Le-man, and Chris North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer*

- Graphics Proceedings of the Visual Analytics Science and Technology 2017*, 24(01), January 2018.
- [321] John Wenskovitch, Michelle Dowling, and Chris North. The cognitive and computational benefits and limitations of clustering for sensemaking. In *Proceedings of the Workshop on Sensemaking in a Senseless World*, CHI'18, 2018.
- [322] John Wenskovitch, Michelle Dowling, Laura Grose, Chris North, Remco Chang, Alex Endert, and David H. Rogers. Machine learning from user interaction for visualization and analytics: A workshop-generated research agenda. In *Proceedings of the IEEE VIS Workshop MLUI 2019: Machine Learning from User Interactions for Visualization and Analytics*, VIS 2019, 2019.
- [323] John Wenskovitch, Michelle Dowling, and Chris North. Simultaneous interaction with dimension reduction and clustering projections. In *Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion*, IUI '19, pages 89–90, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6673-1. doi: 10.1145/3308557.3308718. URL <http://doi.acm.org/10.1145/3308557.3308718>.
- [324] Steve Whittaker and Julia Hirschberg. The character, value, and management of personal paper archives. *ACM Trans. Comput.-Hum. Interact.*, 8(2):150–170, June 2001. ISSN 1073-0516. doi: 10.1145/376929.376932. URL <http://doi.acm.org/10.1145/376929.376932>.
- [325] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, Nov 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70589.

- [326] Margaret Wilson. Six views of embodied cognition. *Psychonomic bulletin & review*, 9(4):625–636, 2002.
- [327] Axel Wismüller, Michel Verleysen, Michael Aupetit, and John Aldo Lee. Recent advances in nonlinear dimensionality reduction, manifold and topological learning. In *ESANN*, 2010.
- [328] Pak C. Wong and Jim Thomas. Guest editors introduction—visual analytics. *IEEE Computer Graphics and Applications*, 24(5):20–21, 24(5), Sep 2004. doi: 10.1109/MCG.2004.39.
- [329] Weng-Keen Wong, Ian Oberst, Shubhomoy Das, Travis Moore, Simone Stumpf, Kevin McIntosh, and Margaret Burnett. End-user feature labeling: A locally-weighted regression approach. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 115–124, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0419-1. doi: 10.1145/1943403.1943423. URL <http://doi.acm.org/10.1145/1943403.1943423>.
- [330] Xuanli Lisa Xie and Gerardo Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(08):841–847, aug 1991. ISSN 0162-8828. doi: 10.1109/34.85677.
- [331] S. Xiong, J. Azimi, and X. Z. Fern. Active learning of constraints for semi-supervised clustering. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):43–54, Jan 2014. ISSN 1041-4347. doi: 10.1109/TKDE.2013.22.
- [332] Rui Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, May 2005. ISSN 1045-9227. doi: 10.1109/TNN.2005.845141.

- [333] M-S Yang. A survey of fuzzy clustering. *Mathematical and Computer modelling*, 18(11):1–16, 1993.
- [334] J. S. Yi, Y. a. Kang, and J. Stasko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, Nov 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.70515.
- [335] Seid Muhie Yimam, Chris Biemann, Ljiljana Majnaric, Šefket Šabanović, and Andreas Holzinger. An adaptive annotation approach for biomedical entity and relation recognition. *Brain informatics*, 3(3):157, 2016.
- [336] Kirsty A Young. Direct from the source: The value of ‘think aloud’ data in understanding learning. *The Journal of Educational Enquiry*, 2005.
- [337] Chong Ho Yu. Exploratory data analysis. *Methods*, 2:131–160, 1977.
- [338] Yisong Yue, Chong Wang, Khalid El-Arini, and Carlos Guestrin. Personalized collaborative clustering. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW ’14, pages 75–84, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2744-2. doi: 10.1145/2566486.2567991. URL <http://doi.acm.org/10.1145/2566486.2567991>.
- [339] Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965.
- [340] Hongyuan Zha, Xiaofeng He, Chris Ding, Ming Gu, and Horst D Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems*, pages 1057–1064, 2002.
- [341] Weizhong Zhao, Qing He, Huifang Ma, and Zhongzhi Shi. Effective semi-supervised document clustering via active learning with instance-level constraints. *Knowledge and*

- Information Systems*, 30(3):569–587, Mar 2012. ISSN 0219-3116. doi: 10.1007/s10115-011-0389-1. URL <https://doi.org/10.1007/s10115-011-0389-1>.
- [342] C. Ziemkiewicz, A. Ottley, R. J. Crouser, A. R. Yauilla, S. L. Su, W. Ribarsky, and R. Chang. How visualization layout relates to locus of control and other personality factors. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1109–1121, July 2013. ISSN 1077-2626. doi: 10.1109/TVCG.2012.180.
- [343] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012.