VISUAL EXPLORATION OF HIGH-DIMENSIONAL SPACES THROUGH IDENTIFICATION, SUMMARIZATION, AND INTERPRETATION OF TWO-DIMENSIONAL PROJECTIONS

by

Shusen Liu

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Computing

School of Computing

The University of Utah

Dec 2016

Copyright © Shusen Liu 2016

All Rights Reserved

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

SUPERVISORY COMMITTEE APPROVAL

of a dissertation submitted by

Shusen Liu

This dissertation has been read by each member of the following supervisory committee and by majority vote has been found to be satisfactory.

Chair: Valerio Pascucci

Dr. Christopher R. Johnson

Dr. Charles Hansen

Dr. Peer-Timo Bremer

Dr. Bei Wang

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

FINAL READING APPROVAL

To the Graduate Council of the University of Utah:

I have read the dissertation of ____________Shusen Liu____________ in its final form and have found that (1) its format, citations, and bibliographic style are consistent and acceptable; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the Supervisory Committee and is ready for submission to The Graduate School.

______________________ ______________________________________

Date

Valerio Pascucci
Chair: Supervisory Committee

Approved for the Major Department

____

Ross T. Whitaker
Chair/Director

Approved for the Graduate Council

____

David B. Kieda
Dean of The Graduate School

ABSTRACT

With the ever-increasing amount of available computing resources and sensing devices, a wide variety of high-dimensional datasets are being produced in numerous fields. The complexity and increasing popularity of these data have led to new challenges and opportunities in visualization.

Since most display devices are limited to communication through two-dimensional (2D) images, many visualization methods rely on 2D projections to express high-dimensional information. Such a reduction of dimension leads to an explosion in the number of 2D representations required to visualize high-dimensional spaces, each giving a glimpse of the high-dimensional information. As a result, one of the most important challenges in visualizing high-dimensional datasets is the automatic filtration and summarization of the large exploration space consisting of all 2D projections. In this dissertation, a new type of algorithm is introduced to reduce the exploration space that identifies a small set of projections that capture the intrinsic structure of high-dimensional data. In addition, a general framework for summarizing the structure of quality measures in the space of all linear 2D projections is presented.

However, identifying the representative or informative projections is only part of the challenge. Due to the high-dimensional nature of these datasets, obtaining insights and arriving at conclusions based solely on 2D representations are limited and prone to error. How to interpret the inaccuracies and resolve the ambiguity in the 2D projections is the other half of the puzzle. This dissertation introduces projection distortion error measures and interactive manipulation schemes that allow the understanding of high-dimensional structures via data manipulation in 2D projections.

CONTENTS

CHAPTERS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This dissertation could not have been accomplished without the help of many whom I would like to thank. I would first like to thank my family, whose endless support made this work possible. I would like to thank my advisor and mentor, Valerio Pascucci, for his continued guidance and encouragement. I would also like to thank the other members of my committee, Timo, Bei, Chuck, and Chris, for their feedback on this work. I would also like to thank the many labmates/collaborators from the Data Analysis group along with my many friends in Salt Lake City.

LIST OF SYMBOLS

nD	n-dimensional
$\mathbb{R}^n$	nD Euclidean space
$\mathbb{R}^p$	nD projective space
$\mathbf{G}$	Grassmannain Manifold
$\mathbf{V}$	Stiefel Manifold
$\mathbf{x}$	column vector (lower case bold Roman letter)
$\mathbf{x}^T$	transpose of a column vector
$(w_1, w_2, ..., w_n)$	row vector with n elements
$\mathbf{w} = (w_1, w_2, ..., w_n)^T$	transpose of a row vector with n element
$\mathbf{M}$	matrix (upper case bold Roman letter)
$(\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_n})$	matrix consists of n column vectors
$det(\mathbf{M})$	determinant of matrix $\mathbf{M}$
V and E	sets of vertices and edges of a graph
$G = \{V, E\}$	graph
f,g, ...	scalar fields
$g(x) = O(f(x))$	$\|f(x)/g(x)\|$ is bounded as $x \to \infty$
$[a, b]$	closed interval from a to b
(a, b)	open interval from a to b
$[a, b)$	an interval includes a but excludes b
Γ	Gamma function $\Gamma(n) = (n-1)!$

Table 0.1: Symbols used.

PART I

INTRODUCTION AND BACKGROUND

CHAPTER 1

MOTIVATION AND CONTRIBUTIONS

With the ever-increasing amount of available computing resources and sensing devices, our ability to collect and generate a wide variety of large, complex datasets continues to grow. From the natural sciences to social sciences, from business to engineering, numerous applications have modeled and studied data with multiple attributes as points in high-dimensional space. For example, in biology, the relationship among genomic microarray have been examined and visualized [3, 4] as high-dimensional data; in air quality research high-dimensional spectrometry data are studied via an interactive dimension reduction visualization framework [5]; in nuclear safety engineering, the failed condition can be analyzed as a high-dimensional scalar function [6, 7], where the input parameters are the domain and the model output is the range; and similarly a high-dimensional scalar function can also be used to study the chemical compounds mixing ratios and the corresponding burning conditions in a combustion simulation [8]. The wide availability and abundance of high-dimensional data and their applications make understanding high-dimensional space one of the important aspects for analyzing complex multiparameter data. However, despite the wide usage, we usually lack intuition and in-depth understanding about high-dimensional space. More importantly, most statistical analysis methods that are often used for analyzing high-dimensional data (e.g., regression analysis) focus on confirming or rejecting certain limited aspects of the data and therefore provide little information regarding the high-dimensional space as a whole or helping develop intuitions for the users. Visualization, as an essential tool for exploratory data analysis, has been proven to be an intuitive and effective alternative for exploring and studying high-dimensional data (e.g., Tukey's early work on projection pursuit [9]). However, there are still many unresolved challenges and new opportunities in visualization for making sense of these high-dimensional data with ever-increasing size and complexity. This dissertation aims to address some of the most prominent challenges when visualizing high-dimensional data through two-dimensional (2D) projections.

1.1 Visualization Challenges

Exploring and visualizing high-dimensional data is essential for making sense of many complex data that have multiple interconnected parameters or properties. However, visualizing such a space is an extremely challenging task.

The physical limitations of the display devices and our visual system prevent the direct display and instantaneous recognition of structures with dimensions higher than two or three. Currently, most effective display devices are 2D in nature. The human eye can perceive the word around us only by the use of a "2D sensor array" (the retina), and even the perception of 3D space is reconstructed through 2D images by the human brain. In addition, our intuitions regarding the spatial relationship and structure are from the 3D space we live in and many of these intuitions fail in high-dimensional space (e.g.,"the curse of dimensionality"; refer to Section 2.1). Understanding even a simple 4D structure may prove to be rather challenging for most people (e.g., Klein bottle). Due to these insurmountable limitations in our ability to understand high-dimensional space, visualization techniques that convey information about the high-dimensional dataset through intermediate visual representations in 2D are becoming increasingly important and necessary. High-dimensional data visualization methods help facilitate the process of transforming high-dimensional information into digestible pieces where the user can gain intuition and understanding of high-dimensional space and the structures within. Here an analogy can be drawn from the famous *blind men and elephant* story [10]. When facing high-dimensional data, we are essentially the blind men who can only try to understand what an elephant is (high-dimensional space) through multiple indirect means (2D visual representations), and these indirect representations could lead to incomplete and misleading information.

Despite many advances have been made in the past decades in both the statistic and visualization communities (for a comprehensive summary, please refer to my survey in [11]), many important challenges remain because of the complexity of high-dimensional datasets and the limitations of the existing approaches

1.1.1 Handling the Enormous Exploration Space

Since both the display device and our visual input channel are limited to 2D, visualization methods rely on intermediate 2D visual representations to express high-dimensional information. As illustrated in Figure 1.1, such a reduction of dimension in a 2D visual representation leads to an expansion of the number of 2D items required for describing the high-dimensional information. As a result, one of the most important challenges in visualizing high-dimensional space is the search, selection, and filtration of the large exploration

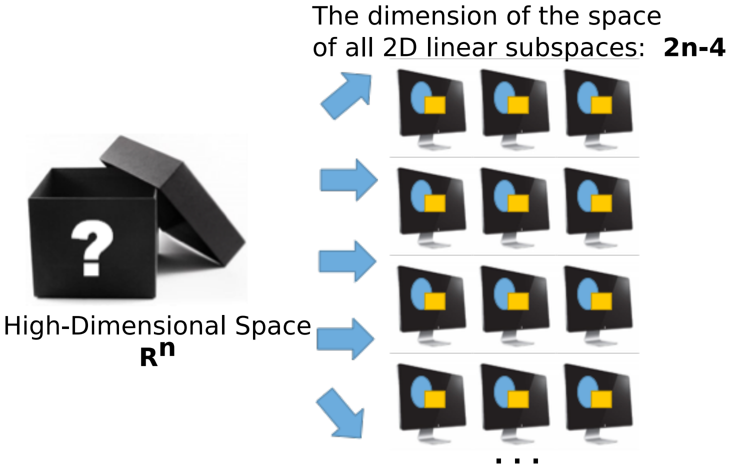

The dimension of the space
of all 2D linear subspaces: **2n-4**
High-Dimensional Space
$\mathbf{R^n}$
. . .

Figure 1.1: The challenge of the extremely large exploration space when visualizing high-dimensional data via 2D representations.

space that consists of all intermediate 2D visual representations. These automatic schemes summarize the useful information, making the exploration of the high-dimensional dataset by a human user, with limited time and energy, possible.

Attributed to the 2D projection's simple construction and innate ability to express data points relationships and structures, it has been one of the most fundamental yet widely used visual representations for visualizing high-dimensional datasets. In this discussion, 2D projection is defined as a mapping from high-dimensional space to 2D space. Such a projection can be axis-aligned projection (usually referred to as a scatterplot), linear projection, or nonlinear projection (manifold learning).

In this dissertation, the 2D projection-based approaches are the focus of the discussion. Let us first take the scatterplot matrix, one of the most widely used high-dimensional data visualization methods, as an example of the challenges of handling enormous exploration space. The scatterplot matrix, or SPLOM, is a collection of scatterplots that allows users to view multiple bivariate relationships simultaneously. As the dimension increases, the number of scatterplots in scatterplot matrix increases quadratically for representing all the bivariate relationships of the dataset. As a result, even for data with dozens of dimensions, the scatterplot matrix will end up with hundreds or even thousands of 2D scatterplots, which will take the user a significant amount of time and energy to explore and analyze. Such a limitation leads to the development of automatic selection and filtration approaches for the 2D scatterplots in a scatterplot matrix. These methods find the interesting or out-of-ordinary scatterplots, based on a given metric, for the user to explore. Scagnostics [1] is one such attempts. It provides a set of nine measures capturing properties such as outliers, shape, trend, and density in scatterplot matrix for identifying "interesting" plots (scatterplots that agree with certain patterns). These measures for identifying interesting patterns in high-dimensional datasets are usually referred to as quality metrics. We can rank the projections based on the quality metrics scores (e.g., Scagnostics [1], rank-by-feature framework [12]), which provide an indication of whether a scatterplot should be closely examined by a human user.

Although using a flat ranking allows one to discard "bad" projections, it may not be ideal for selecting "good" ones. Firstly, there is no guarantee that the "interesting" and "informative" projections are included in the candidate set to begin with. For the scatterplot matrix, all the projections are axis-aligned and therefore include only a fraction of all possible 2D linear projections of the high-dimensional dataset. Secondly, the ranking does not give a clear indication of how the selected projections might be related. For example, does the second "best" projection have a high ranking simply because it is very similar to the projection ranked as number one? A more desirable situation is where the user is presented with projections that complement each other and highlight different aspects of the dataset. As a result, ranking based solely on the quality metric is not enough, and the method needs to make more intelligent decisions and identify "locally" best projections that together paint a more comprehensive picture of the dataset.

Moreover, the scatterplot matrix captured only the bivariate relationship among different dimensions. Implicitly, it also treats all dimension as equal. However, for many datasets, the relationships between dimensions are not uniform. The closely related dimensions can and should be grouped into clusters, where the noise from the unrelated dimensions is removed, for subsequent analysis. The methods that find clusters within the subset of dimensions are usually referred to as subspace clustering methods (e.g., ENCLUS [13], SURFING [14]), initially developed in the data mining and knowledge discovery community. Subspace clustering can also be considered as a class of methods for reducing the large exploration space. By focusing on individual clusters, where only a small subset of dimension is considered, the exploration and analysis process is simplified. The subspace clustering methods have been adopted [15, 16] for visualizing high-dimensional space. These methods identify the intricate relationship among dimensions, introducing some very interesting exploration strategies for high-dimensional datasets and can be particularly effective when the dimensions are not tightly coupled.

However, some drawbacks remain. Firstly, these subspace clustering methods group subsets of dimensions and therefore captures only the axis-aligned properties during the clustering process. Secondly, these methods will fail when applied to data where dimensions are close related, such as the face images dataset where each dimension corresponds to a pixel value. Thirdly, these methods are prone to generate a large number of candidate clusters, where an automatic search and filtering operation is again necessary for the exploration of the results, which defeats the initial purpose of exploration space reduction.

In this dissertation, a new type of algorithm is introduced to identify the projections that capture the intrinsic structure of high-dimensional data, which drastically reduces the exploration space. In addition, a general framework for summarizing the structure of quality metrics in the space of all linear projections is presented, which provides a more intelligent projection-selection approach compared to ranking the quality metrics scores directly.

1.1.2 Interpreting the 2D Representations

Identify the "interesting" or "informative" projections of the high-dimensional dataset is only part of the challenge. How a user can analyze, interpret, and understand these selected 2D representations of high-dimensional data is the other half of the puzzle. As illustrated in Figure 1.2, when 2D projections are generated for visualizing high-dimensional space, the information loss is unavoidable. Therefore, obtaining insights and arriving at conclusions based solely on the 2D representation are limited and prone to error.

For 2D projections, the error originates from the inability to express the complex high-dimensional relationship in the limited 2D space (e.g., further apart points in high-dimensional space may be projected onto the same neighborhood in 2D or vice versa). Such error is often referred as distortion error. The objective function of dimensionality reduction methods (e.g., principal component analysis, linear discriminate analysis) can provide some indication regarding how well a given projection preserves the high-dimensional features with respect to the formulation of the objectives. However, the objective functions or other types of global measures produce only one number, which is woefully inadequate for

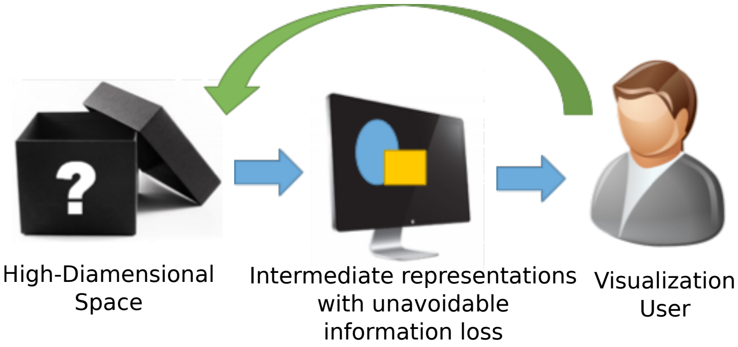

High-Diamensional Space
Intermediate representations with unavoidable information loss
Visualization User

Figure 1.2: The challenge of interpreting the intermediate 2D representation where unavoidable information loss occurs.

capturing the variations within each projection. For example, a projection may capture part of the high-dimensional structure extremely well, but completely falter on the rest. A per-point estimation regarding how well localized features are preserved is extremely valuable for interpreting the 2D projection results. In this dissertation, the concept of per-point quality measure is extended to the general measures that are applicable to various types of projections, and type-specific ones that correspond to how the projections are generated (including linear and nonlinear dimension reduction methods).

While the per-point distortion measures help to identify where the errors or inaccuracies are, it does not aid in explaining why such errors exist. To overcome such limitations, identifying a link between the 2D space and the original high-dimensional space is essential for an in-depth understanding of the dataset. In this dissertation, a distortion-guided manipulation scheme is introduced to address the “why” question by allowing the user to manipulate high-dimensional structures in 2D while providing interactive feedback through per-point distortion measures.

1.2 High-Dimensional Data Definition and Classification

Before further discussion, providing a definition for high-dimensional data in the context of this dissertation is necessary. High-dimensional datasets can be defined through the perspective of the *domain* and *range* of a function. The domain attributes correspond to the coordinates in an abstract space; the range attributes are the function values defined on the domain.

As illustrated in Figure 1.3, such an interpretation provides a unified view of several related but different types of datasets. If the dimension of either the domain or range is higher than three, this dataset is considered as high-dimensional. For example, multivariate volumetric dataset that often seen in various scientific simulations is one type of high-dimensional datasets, where the dimension of the domain is three and the dimension of the range is more than three.

Multidimensional is usually used to describe the dataset with a modest dimension that is not significantly higher than three. High-dimensional, on the other hand, suggesting a larger dimension count than multidimensional. However, there are no obvious criteria to determine exactly how many dimensions can be considered high-dimensional and how many dimensions are just multidimensional. In addition, for datasets with dimensions higher than three, the methods for visualizing them are usually fundamentally different from the methods for 2D or 3D datasets. Moreover, for many datasets, despite having large data dimensions, the dataset’s intrinsic dimension can be surprisingly low. For example, an image from the face

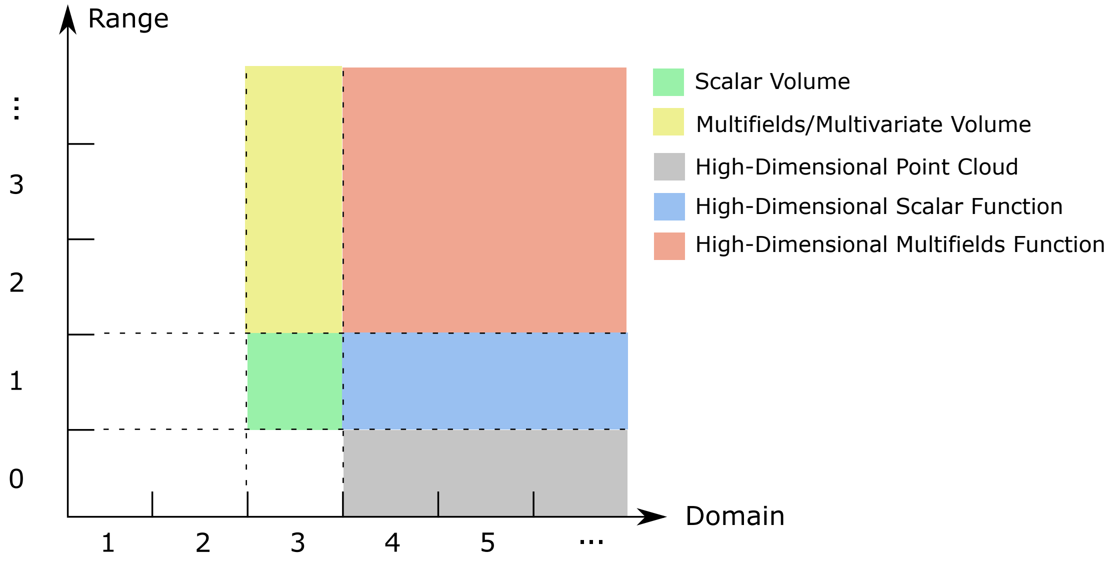

Range
⋮
3
2
1
0
Scalar Volume
Multifields/Multivariate Volume
High-Dimensional Point Cloud
High-Dimensional Scalar Function
High-Dimensional Multifields Function
1
2
3
4
5
⋯
Domain

Figure 1.3: Illustration of the range domain interpretation of a dataset. It provides a unified view of various different but related types of data, including variations of high-dimensional dataset.

images database (A face image may contain thousands of pixels. If each pixel describes one dimension, each image will represent one point in a very high-dimensional space [17]). However, the relationship between these images (high-dimensional points) can be captured by a much lower dimensional space (its intrinsic dimension). These observations further blur the line between the concepts of multidimensional and high-dimensional. Therefore, in this dissertation, the terms multidimensional and high-dimensional are not strictly separated. High-dimensional is used in a more generalized sense describing both multidimensional and high-dimensional datasets.

1.3 Identify Informative 2D Projections

In light of the previous discussion, one of the major challenges in visualizing high-dimensional data is handling the enormous exploration space, specifically, how to automatically search for and select a set of informative 2D projections that best captures the properties and features of the high-dimensional data.

2D projection generated by various high-dimensional visualization methods can be roughly divided into three categories: bivariate scatterplot, linear projection, and nonlinear projection. The bivariate scatterplot (as in a scatterplot matrix) is easy to understand, since its axes directly correspond to the original dimensions. Linear projections, where the axes of the plots are the linear combination of existing dimensions, are more challenging to understand but at the same time are more likely to capture important structural information (compared to the scatterplot's axis-aligned projections). As for nonlinear projections, even nonlinear manifold structures can be learned, but the axis of the resulting projection loses its meaning. Therefore, as illustrated in Figure 1.4, delicate trade-offs exist between the interpretability of the axis and the potential for capturing intrinsic structure in the data. To strike a balance between capturing the intrinsic structures and generating interpretable results, the linear projection is targeted for the proposed projection-finding techniques.

Many existing projection-finding algorithms [1, 18, 19, 20] rely on various quality measures to rank the potentially important projections from a set of candidate samples. There are several issues with these approaches when samples are filtered by quality measures. First, the initial candidate set heavily influences the outcome of the result. If the candidate set includes only axis-aligned projections, only bivariate relationships can be discovered. As a result, potentially important projections may not be selected simply due to small or limited candidates. Second, most quality measures are designed to highlight the out-of-ordinary configurations (e.g., deviate from a Gaussian distribution for the project pursuit index [9]) without considering the underlying data's intrinsic structure. Such a process indicates that the identified projections may not correspond to the important structural information, but instead highlight the less significant structure that happens to fit the patterns that the particular quality measures are seeking, thereby leading to potentially misleading representative projections. Approaches based on subspace clustering methods [15, 4, 21] that find clusters in the subsets of dimensions introduce interesting alternatives for identifying informative 2D projections. However, the axis-aligned constraint and nonuniform assumption among the dimensions limited their use. Instead of focusing on the subset of dimensions, identifying linear subspaces (non-axis-aligned) provides a more flexible alternative.

Recently, advances have been made in the machine learning community for performing non-axis-aligned subspace clustering [22]. Instead of grouping dimensions, the points are

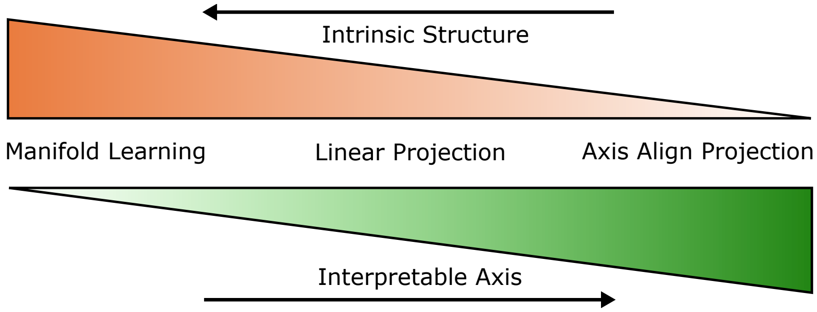


Figure 1.4: Interpretable axis vs. intrinsic structure; the trade-off between different types of 2D projections.

grouped together for sharing similar linear subspaces (as illustrated in Figure 1.5). These methods help to decompose the high-dimensional space into multiple smaller but simpler regions, where a lower dimension space is sufficient to capture the interpoint relationship. In this dissertation, these non-axis-aligned subspace clustering methods [22, 23, 24] are introduced as the basis for the new class of projection selection methods for identifying the informative linear projection of a high-dimensional dataset.

However, finding the 2D representations alone does not necessarily help the user understand the dataset as a whole. A clear understanding of the relationships among these 2D representations is the key. In this research, by introducing a view navigation graph that provides flexible navigations among these selected 2D projections from subspace analysis, intuitive exploration of the informative projections and their relationships in the high-dimensional space is achieved.

1.4 Summarize the Space of 2D Projections

As discussed in the previous section, one of the fundamental challenges of visualizing high-dimensional space is to identify informative 2D projections that capture the intrinsic structure of the data. However, even for datasets with moderate dimensions, exploring all possible axis-aligned projections, let alone all linear ones, becomes impractical. Therefore, as discussed earlier, a common strategy is to search through a large number of potentially interesting projections and select a small set based on a ranking of quality measures computed from the projections.

However, few techniques explicitly consider diversity when choosing representative projections. As a result, multiple highly ranked but redundant (similar) projections may be

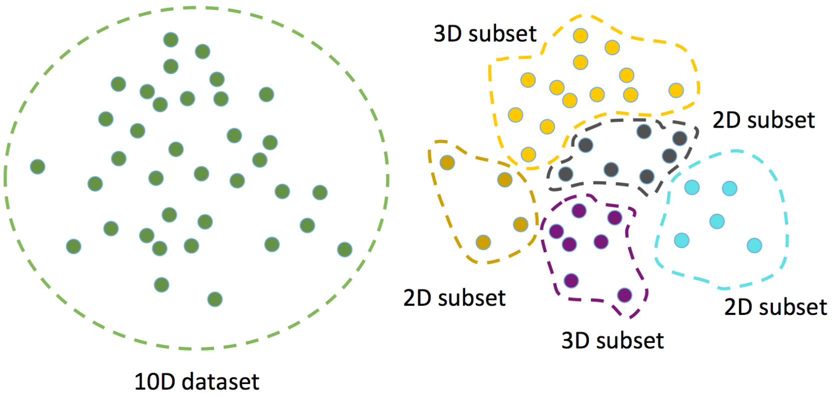


Figure 1.5: The intuition behind subspace analysis. A given high-dimensional space can be decomposed into multiple lower dimensional linear subspaces.

selected. At the same time, lower ranked ones are discarded even though they may contain complementary information. Simply increasing the number of selected projections does not mitigate the diversity issue, but on the contrary, may increase the likelihood of selecting multiple similar projections with large quality measure values.

On the other hand, each quality measure is designed to capture some aspects of the data, yet little is known regarding the properties of the measure. For example, understanding the *smoothness* of a measure and the distribution of its *local maxima* are crucial in choosing the right representative projections. In particular, through the experiments carried out in this dissertation, many quality measures have been found containing only a single maxima globally that may not be suitable for finding multiple projections.

In this dissertation, the *Grassmannian Atlas*, a new framework to analyze, compare, and explore the space of all linear projections based on different quality measures, is introduced. Rather than working with a few selected projections, the space of linear projections is modeled by the so-called *Grassmannian* [25], which abstracts the space of linear subspaces in a data-independent manner and compensates for affine transformations of the projections. The Grassmannian is approximated by connecting a set of *sampled* points (each corresponding to a subspace) on the manifold with a neighborhood graph based on well-defined geodesics. Then, a given quality measure is analyzed as a scalar function defined on the Grassmannian and the notion of *locally optimal* projections is introduced: the local maxima of the quality measure that are robust to small perturbations of the function. Consequently, using tools from scalar field topology, a topological skeleton can be extracted that describes the number, locations, and relationships among optimal projections. Such a skeleton can be simplified and visualized via the topological spines [26]. The topological spines provide a 2D multiresolution representation of the otherwise high-dimensional structure, which leads to a visual map for exploring the space of projections in an intuitive manner.

In addition, by introducing the concept of the Grassmannian (the space of all n-dimensional linear subspaces), this dissertation establishes a unified framework for exploring linear projection of high-dimensional datasets and providing the theoretical foundation for studying the relationships (e.g., distances) among linear projections and linear subspaces.

1.5 Interpret 2D Projections via Manipulation

Despite the enormous efforts put into methods generating informative 2D projections, limited state-of-the-art works have been dedicated to the endeavor of interpreting and making sense of these 2D projections. Obtaining insights and arriving at conclusions based solely on the 2D projections are limited and prone to error. Therefore, how to make sense and interpret these 2D representations in the context of high-dimensional space is as crucial as finding the projections.

To address the problem of misleading information in a 2D projection, effectively conveying the inaccuracies is essential. The inaccuracies in the projection can be evaluated from two perspectives: first, a global measure of the absolute magnitude of the error, which addresses the question: Is the projection totally misleading? Second is a per-point estimation of the error, which addresses the question: Should I trust a given point in the projection? Here the per-point error is particular interesting, as it provides the user with adequate information to determine whether the inconsistency in the projection is likely to have been introduced by the dimension reduction process. This dissertation derives both the global and per-point error measures for linear projections and various nonlinear dimensionality reduction methods, which addresses the challenges of identifying misleading projections and the misleading areas within a projection.

For a given projection, the per-point error measures answer the question of where the inaccurate areas are. However, relying on the measure alone, we still cannot answer the question of why some of the highly distorted areas exist. Interactivity plays an extremely important role in visualization. For example, Brown et al. [27] introduce the distance-function-learning concept, where a new distance metric can be calculated from the manipulation of point layouts by an expert user. Such an interactive manipulation scheme allows users' knowledge to be incorporated into the algorithm. In this research, a projection manipulation scheme is introduced. It extrapolates the manipulation applied in the 2D space to the original high-dimensional dataset and reflects the changes in 2D via the on-the-fly update of per-point error measures. By utilizing interactive exploration and manipulation of projection results, a deeper understanding of these projections is made possible, which leads to new intuition and insights of the high-dimensional dataset.

One fundamental challenge when manipulating projected points in 2D is the lack of high-dimensional structure information. Due to the constraints and limitations of 2D space, common interaction tools such as lasso or box selection may select points that belong to far away high-dimensional neighborhoods, which introduce more inaccuracies rather than helping resolve the ambiguity. A meaningful data manipulations (e.g., data movement and data deletion) in the visual space should be structure-driven, that is, the selected points should respect certain structures of the original high-dimensional data. In order to overcome this obstacle, structural context, computed from hierarchical clusterings, is imposed onto the embeddings as a multiresolution skeleton, which serves as a structural abstraction of the data at multiple scales and handles for manipulation.

1.6 Dissertation Contributions

In summary, the research done as part of this dissertation has led to the development of a self-contained framework for visualizing high-dimensional space. The framework not only helps users identify 2D projections that reveal intrinsic structures of the dataset and summarize the space of all linear projections, but also provides the tools that aid in the interpretation of the 2D projections in connection with the original high-dimensional space. In addition, all the proposed techniques are readily available as components that work together in a unified software system.

The key contributions are itemized as follows:

- **Identify Informative 2D Projections** (PART II, Chapters 4, 5, 6)

  First, a new type of algorithm [28] is introduced for identifying the linear projections that capture the intrinsic structure of high-dimensional data, which drastically reduces the exploration space. The method identifies the informative linear projections by utilizing subspace analysis, reveal and summarize the relationship among these projections through a view navigation graph. In addition, since the proposed technique provides an intuitive interface for exploring high-dimensional space from multiple perspectives (2D projections), the technique is extended to multivariate volume visualization [29] for designing multivariate transfer functions. Finally, subspace analysis is also utilized for making sense of analogy relationships in the word embedding space.

- **Summarize the Space of 2D Projections** (PART III, Chapter 7)

  Second, a general framework [30] for summarizing the space of all linear projections of high-dimensional data is presented. Chapter 7 introduces a unified framework for working with linear subspaces and understanding their relationships (with well-defined distance metrics). The proposed work, the Grassmannian Atlas, captures the global structures of quality measures via topological data analysis in the space of all 2D subspaces, which enables a systematic exploration of many complementary projections (local maxima) and also provides new insights into the properties of existing quality measures.

- **Interpret 2D Projections Via Manipulation** (PART IV, Chapters 8, 9)

  Finally, a projection manipulation scheme [31] is introduced to facilitates the understanding of high-dimensional data via manipulation of its 2D projections (linear and nonlinear). The structural abstractions obtained through hierarchical clusterings allow multiscale data manipulations, even with hidden or occluded data points in

2D. Combining interactive data manipulations in the 2D projection with on-the-fly updates of distortion measures provides new insights regarding structural relations among different parts of the data.

Novel visualization techniques, without a proper delivery mechanism, can not reach their intended audience. The design and implementation of the visualization system have a profound impact on the usability of the proposed method. In this dissertation, conscious efforts have been made into producing a complete, self-contained, and usable software framework, namely *DataExplorerHD* (`http://goo.gl/FnnOKs`). The *DataExplorerHD* includes all the above-mentioned techniques and designs to be flexible and extensible to meet future demands.

1.7 Dissertation Structure

The structure of the remaining chapters in this dissertation is outlined below:

- **Chapter 2:** discusses the background and definitions that are the foundation of this dissertation.
- **Chapter 3:** covers the related works in the field of high-dimensional data visualization relevant to the discussion in this dissertation.
- **Chapter 4:** introduces the subspace analysis for identifying informative linear 2D projections, summarizing their relationship and navigating among these projections.
- **Chapter 5:** introduces the application of subspace analysis approach for designing transfer function and visualizing multivariate volume data.
- **Chapter 6:** finding informative 2D projections for making sense of the analogy relationships in the high-dimensional word embedding space.
- **Chapter 7:** discusses a unified framework, the *Grassmannian Atlas*, for exploring the space of all linear subspaces.
- **Chapter 8:** explores the distortion error measures for highlighting inaccuracies in 2D projections.
- **Chapter 9:** presents a distortion measures guided interactive manipulation scheme that aid in the interpretation of projection results.
- **Chapter 10:** concludes the dissertation and discusses potential future directions.

CHAPTER 2

BACKGROUND AND DEFINITIONS

In this section, some important properties of high-dimensional data are discussed. Mathematical definitions that provides foundations for understanding the rest of the dissertation are covered.

2.1 Properties of High-Dimensional Space

Naturally, for data with multiple attributes, each entry (record) can be associated with a point in the high-dimensional space spanned by the attributes. Despite the simple construction, high-dimensional space can contradict intuitions obtained through our daily lives. To get a sense of the behavior of high-dimensional space, the properties of simple high-dimensional geometries are studied first. Then, the distance metrics in high-dimensional space are discussed. Finally, the curse of dimensionality and its implications regarding high-dimensional data visualization are examined.

2.1.1 Simple High-Dimensional Geometry

As a first step for understanding high-dimensional space, let us take a look at some simple geometry, namely cube and sphere, and how they behave in high-dimensional space.

A hypercube is one of the simplest geometries in high-dimensional Euclidean space. In the following discussion Euclidean space is assumed. it is a generalization of the cube to n-dimensional space. The volume of the hypercube in n-dimensional space is $V_{cube} = r^n$, where r is the edge of the cube. For a unit hypercube (hypercube where the length of the side is 1), the relationship between its properties (e.g., diagonal length, vertex/corner count) and dimension is illustrated in Table 2.1. Since the edge of the unit cube is always 1, as the dimension increases the volume of the unit cube stays at 1. However, the diagonal of the cube is $d_{diag} = \sqrt{n}$, which lead to an interesting observation about the cube in high-dimensional space. Based on the definition of d_{diag}, we can conclude that as $n \to \infty$ the diagonal d_{diag} will also approach infinity.

Table 2.1: Unit hypercube properties as the dimension increases.

dimension	V_{cube}	vertex/corner count	diagonal length
1	1	2	1
2	1	4	$\sqrt{2}$
3	1	8	$\sqrt{3}$
4	1	16	$\sqrt{4}$
5	1	32	$\sqrt{5}$
6	1	64	$\sqrt{6}$
n	1	2^n	$\sqrt{n}$
∞	1	∞	∞

Intuitively, we can imagine the unit hypercube in high-dimensional space is somewhat "spiky", where the tip of the "spike" is $\frac{\sqrt{n}}{2}$ away from the center of the hypercube (illustrated in Figure 2.1).

Next, let us take a look at the hypersphere (the generalization of the sphere in n-dimensional space), which has a very different behavior compared to the hypercube. The volume of a hypersphere in n-dimensional space can be defined as follows: $V_{sphere} = \frac{\pi^{n/2} r^n}{n\Gamma(n/2+1)}$, where r is the radius of the sphere, and Γ is the *Gamma* function $\Gamma(n) = (n-1)!$. The formulation of the hypersphere volume can be derived by direct integration in the spherical coordinate. By examining the variation of the volume of the unit hypersphere as the dimension increases (see Table 2.2), we can see that the volume first increases and then decreases. The volume reaches a maximum at around the dimension number 5. However, the number 5 does not hold any particular meaning as the peak of volume will vary according

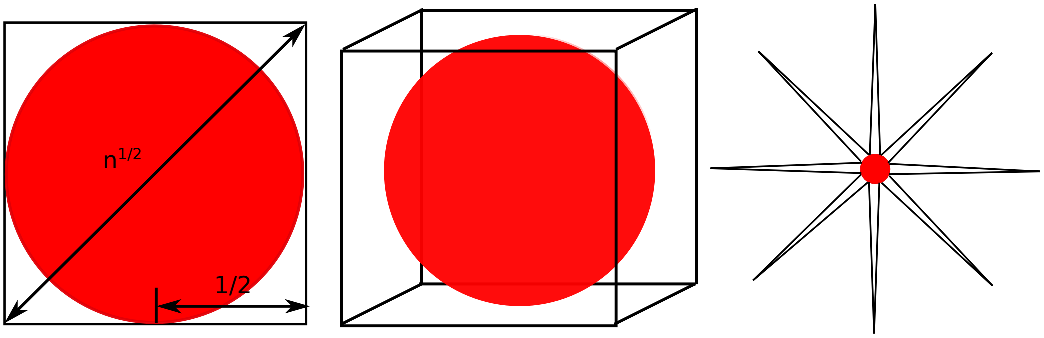


Figure 2.1: Illustration of the ratio between a unit hypercube and a hypersphere with a radius of $\frac{1}{2}$ in different dimensions. As dimension n increases, the diagonal of the cube $d_v = n^{\frac{1}{2}}$ also increases, while the volume of unit cube remains 1. The concept of a high-dimensional unit hypercube is illustrated in the right image, where the diagonals of the hypercube become much larger than 1 and form "spike"-like structure. Finally, $\lim_{n\to\infty} \frac{V_{sphere(r=1/2)}}{V_{cube}} = 0$, where V_{cube} is the volume of the unit hypercube and $V_{sphere(r=1/2)}$ is the volume of the hypersphere with a radius of $\frac{1}{2}$.

Table 2.2: Hypersphere properties as the dimension increases.

dimension	unit sphere $V_{sphere(r=1)}$	$V_{sphere(r=1/2)}$	$\frac{V_{sphere(r=1/2)}}{V_{cube}}$
2	3.14159	0.78540	0.78540
3	4.18879	0.52360	0.52360
4	4.93480	0.30843	0.30843
5	5.26379	0.16449	0.16449
6	5.16771	0.080746	0.080746
n	$\frac{\pi^{n/2}}{n\Gamma(n/2+1)}$	$\frac{1/2^n \pi^{n/2}}{n\Gamma(n/2+1)}$	$\frac{1/2^n \pi^{n/2}}{n\Gamma(n/2+1)}$
∞	0	0	0

to the radius, as illustrated by changing the radius to $\frac{1}{2}$ (see Table 2.2). As the dimension approaching infinity, the volume of the hypersphere will also approach zero. As a result, as illustrated in Figure 2.1 and Table 2.2, the ratio between $V_{sphere(r=1/2)}$ and V_{cube} approaches zero as the dimension approaches infinity: $\lim_{n\to\infty} \frac{V_{sphere(r=1/2)}}{V_{cube}} = 0$, where V_{cube} is the volume of the unit hypercube and $V_{sphere(r=1/2)}$ is the volume of the hypersphere with a radius of $\frac{1}{2}$.

These bizarre behaviors of the simplest geometric structure signify the vast difference between the high-dimensional structures and their low-dimensional counterparts. As a result, when studying high-dimensional space, we need to reevaluation our intuition regarding common measures, even the simplest ones, such as distance and volume. In the following section, the behavior of the distance metric, which has a profound impact on understanding high-dimensional space, is investigated.

2.1.2 Distance Metric

When studying a geometric space, understanding the distance metrics (defined in such a space) and their behavior is essential. In our daily lives, we have developed intuition regarding distance in 2D or 3D space. However, in high-dimensional space, as indicated by the discussion in Section 2.1.1, the distance will very likely not behave as we expect.

Considering an n-dimensional dataset, in which each dimension has uniformly distributed values between [-1, 1]. If a value is between [-0.2, 0.2], it can be considered as close to zero in this example. For an nD data point, the probability of it being close to origin is $(\frac{0.4}{2})^n = (\frac{1}{5})^n$. As the dimension increases, the probability of a given point close to the origin $((\frac{1}{5})^n)$ will decrease exponentially. Now, consider two data points following the same setup. We consider the attribute values to be similar only if their difference is less than 0.4. For these two points to be close, each dimension needs to have similar values. As a result, the probability of two given nD points to be considered nearby is $(\frac{2}{5})^n$. Such observations indicate all points in a high-dimensional space will likely be far away from each other and also far away from the origin, provided the values in each dimension are uniformly distributed. The sparsity of the high-dimensional space can also be understood by a space filling metaphor. For a 1D domain $[0, 10]$, the number of unit 1D hypercube needed to fill the space is 10. As dimension increases, the number of unit hypercubes required to fill the domain increases exponentially (for an nD domain, the number is 10^n). As a result, in many high-dimensional datasets, the data points appear to be far away from each other. In addition, the relative values of the standard distance measure, such as Euclidean distance ($d_{L2} = \sqrt{||p-q||^2}$, where p and q are nD vectors), become less meaningful as dimension increases.

Since most high-dimensional points will be far away from each other while pointing to different directions from the origin, one possible alternative to alleviate the distance metric problem is to view distance as the angle between the direction defined by each point and the origin. Essentially, instead of viewing the space spanned by the data attribute as a Cartesian coordinate, we can view it as a vector space. The distance defined by such an interpretation is referred to as the cosine distance, is computed as follows:

$$d_{cosine} = cos\theta = \frac{p \cdot q}{|p| \cdot |q|}$$

where p and q are nD vectors.

2.1.3 Curse of Dimensionality

From the previous discussion regarding the distance measure in high-dimensional space, we learned as the data dimension increases, for uniformly distributed data points the inter-point distances become less meaningful. In addition, for the same amount of data points, as the dimension increases, space will become more and more sparse. These fundamental properties make designing effective algorithms in high-dimensional space extremely challenging. Often these algorithms have an exponential time complexity in respect to the dimension. For example, spatial index methods such as KD-tree, which work extremely well for 3D data, cannot easily scale to high-dimensional space. In machine learning, most classifiers, such as the k-nearest neighbor, will also become less effective as the dimension increases. This phenomenon is usually referred to as the *curse of dimensionality.*

The curse of dimensionality directly impacts most high-dimensional data visualization methods. Many visualization techniques rely on multiple low-dimensional representations that can grow from quadric to exponential with respect to dimension. The implications of the curse of dimensionality for high-dimensional data visualization are multifold. On one hand, the visualization algorithm that focuses on subspace (linear subspace or axis-aligned subspace), where the high-dimensional space is decomposed into distinct subsets, can greatly mitigate the impact of dimension in terms of the complexity of visual representations. On the other hand, the input data dimension is put under additional scrutiny. Since the cost of large dimension is high, we may ask ourselves: Do we really need all these dimensions to capture the information presented in the data? For some data, the interpoint relationship can be captured with a much lower-dimensional space (e.g., an image data where each pixel corresponding to a dimension contains a large amount of redundant information). Therefore, identifying the intrinsic dimension of the data, and reducing its dimension before visualizing may produce a more desirable result.

2.2 Linear Projection and Linear Subspace

Projection, in general, defines a mapping of a set into a subset. In the context of this dissertation, projection refer to the process by which high-dimensional data are mapped onto a lower-dimensional space. More specifically, a linear projection, defined as an $n \times m$ linear transformation matrix $F = (\mathbf{w_1}, ..., \mathbf{w_m})$, maps a nD vector space into a mD vector space. The mD vector space is spanned by the orthonormal basis $\mathbf{w_1}$, ... ,$\mathbf{w_m}$. Here, n is the input data dimension and m is the target dimension ($n > m$). Linear projection is one of the most widely used methods for generating a 2D representation of high-dimensional data. It provides a good tradeoff between the expressiveness and the ease of interpretation.

For a given nD data, a linear projection defines an orthonormal basis and produces one projected image with explicit coordinates. Linear subspace, on the other hand, does not explicitly define the basis, as long as the given projection basis spans the same subspace. As illustrated in Figure 2.2, the projected images (a) and (b) are produced from different linear projections. However, for visualization purposes, the projected image contains the exact same pattern. In other words, linear projections are not rotation and direction invariant, i.e., different projections can essentially produce the same projected image albeit with a different orientation. Therefore, the notion of linear subspaces can be used to more accurately capture the different projected image configurations.

2.3 Grassmannian as a Unified Framework

In the previous section, the distinction between linear projection and linear subspace was discussed. For visualization purposes, linear subspace is more suitable to capture the difference among projected image configurations. In this section, the space of linear subspace, formally defined as the Grassmannian, is examined. By adopting the Grassmannian as the

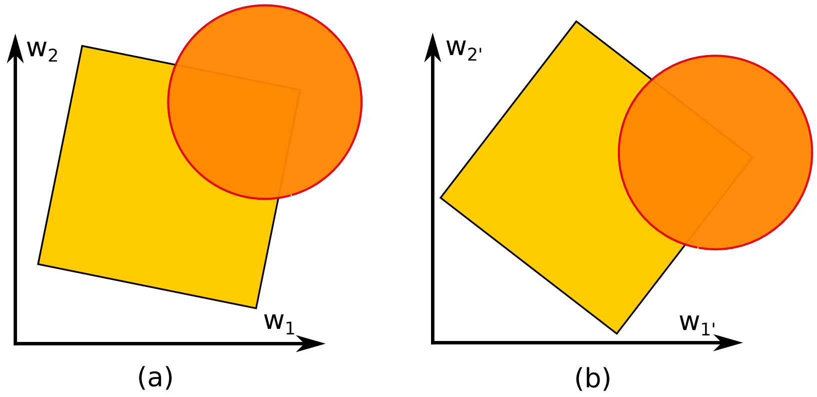


Figure 2.2: An illustration of different linear projections that capture the same linear subspace.

theoretical foundation, the relationship between linear subspaces can be studied under a unified framework.

2.3.1 The Space of all Linear Subspaces

The geometry of non-Euclidean spaces gives rise to the notion of manifolds, and in particular the space of linear subspaces can be effectively described by a Grassmann manifold (or Grassmannian). The Grassmannian, $Gr(k,n)$, is a set of $k-$dimensional linear subspaces of $\mathbb{R}^n$, where each subspace maps to a unique point on the manifold [25]. An important characteristic of the Grassmannian is that there is no unique order or choice of basis (invariant to rotation) for an element on the manifold. Furthermore, it inherits a Riemannian metric from the Euclidean metric on $\mathbb{R}^{n\times k}$, and hence induces a geodesic distance for comparing different subspaces. For a $k-$dimensional subspace W, let $\{b_i\}_{i=1}^k$ denote a set of orthonormal basis vectors that span W. The basis matrix $\mathbf{B} \in \mathbf{R}^{n\times k}$ represents a linear projection to the subspace. By column-reducing the first $k \times k$ block to an identity matrix, the last $(n-k)\times k$ block specifies the coordinates of a given subspace W on $Gr(k,n)$. As a result, the dimension of a smooth Grassmannian $Gr(n,k)$ is given as $k \times (n-k)$. Since we are interested only in the case of 2D subspaces, the resulting dimension is $2 \times (n-2)$.

2.3.2 Distance Between Linear Subspaces

By introducing the concept of the Grassmannian, the distance between linear subspaces can be described by a set of clearly defined metrics.

Given two points on a Grassmannian, represented by their orthonormal bases, $\mathbf{A}$ and $\mathbf{B}$ of size $n \times k$, the distance measured along the geodesic is the *Grassmann distance.*

The geodesic distance can be computed by decomposing $\mathbf{A}^T\mathbf{B}$ using its SVD (Singular Value Decomposition) and obtaining $\sum_{i=1}^{k}\left(\theta_i^2\right)^{\frac{1}{2}}$. Here, θ_i denotes a principal angle and is obtained as $\cos^{-1}\sigma_i$, where σ_i is the corresponding singular value.

Other commonly used distance metrics defined on the Grassmannian beside the *Grassmann distance* include:

- Asimov distance: $\theta_k = \cos^{-1}\|\mathbf{A}^T\mathbf{B}\|$
- Binet-Cauchy distance: $(1-\prod_{i=1}^{k}\cos^2\theta_i)^{1/2} = (1-det(\mathbf{A}^T\mathbf{A})^2)^{1/2}$
- Cordinal distance: $(\sum_{i=1}^{k}\sin^2\theta_i)^{1/2} = \frac{1}{\sqrt{2}}\|\mathbf{A}^T\mathbf{A}-\mathbf{B}^T\mathbf{B}\|_F$
- Projection distance: $\sin\theta_k = \|\mathbf{A}^T\mathbf{A}-\mathbf{B}^T\mathbf{B}\|_2$

For computation efficiency, the Chordal distance [32]($(\sum_{i=1}^{k}\sin^2\theta_i)^{1/2}$) is widely adopted in lieu of the true geodesic distance, which can be computed directly from a pair of orthonormal bases as $\frac{1}{\sqrt{2}}\left\|\mathbf{A}\mathbf{A}^T-\mathbf{B}\mathbf{B}^T\right\|_F$.

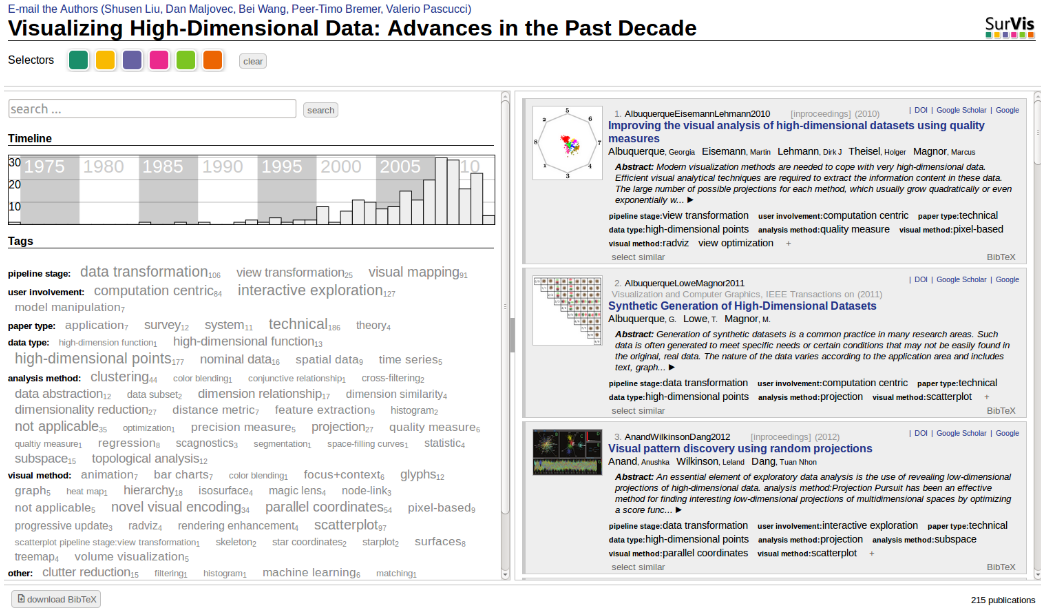
CHAPTER 3

RELATED WORK

Numerous techniques have been proposed for visualizing high-dimensional space in both the visualization and statistics community in the past decades. In order to gain a deeper understanding of the topic, I have conducted, in collaboration with Dan Maljovec and colleagues, an extensive survey (published in [33]) of the literature that focuses on high-dimensional data visualization techniques. In addition, I maintain a website hosting an up-to-date online library (Figure 3.1, `http://goo.gl/hN7G7x`) of related references for visualizing high-dimensional data (more than 200 papers), which allows easy querying and filtering of existing works. In this chapter, a small subset of all survey works is included to provide context and background for the dissertation.

3.1 Scatterplot Matrix

A scatterplot matrix (see Figure 3.2), or SPLOM, is a collection of bivariate scatterplots that allows users to view multiple bivariate relationships simultaneously. One of the



E-mail the Authors (Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, Valerio Pascucci)

Visualizing High-Dimensional Data: Advances in the Past Decade

Tags

pipeline stage: data transformation$_{106}$ view transformation$_{25}$ visual mapping$_{91}$

user involvement: computation centric$_{84}$ interactive exploration$_{127}$ model manipulation$_{7}$

paper type: application$_{7}$ survey$_{12}$ system$_{11}$ technical$_{186}$ theory$_{4}$

data type: high-dimension function$_{1}$ high-dimensional function$_{13}$ high-dimensional points$_{177}$ nominal data$_{16}$ spatial data$_{9}$ time series$_{5}$

analysis method: clustering$_{44}$ color blending$_{1}$ conjunctive relationship$_{1}$ cross-filtering$_{2}$ data abstraction$_{12}$ data subset$_{2}$ dimension relationship$_{17}$ dimension similarity$_{4}$ dimensionality reduction$_{27}$ distance metric$_{7}$ feature extraction$_{9}$ histogram$_{2}$ not applicable$_{35}$ optimization$_{1}$ precision measure$_{5}$ projection$_{27}$ quality measure$_{6}$ qualtiy measure$_{1}$ regression$_{8}$ scagnostics$_{3}$ segmentation$_{1}$ space-filling curves$_{1}$ statistic$_{4}$ subspace$_{15}$ topological analysis$_{12}$

visual method: animation$_{7}$ bar charts$_{7}$ color blending$_{1}$ focus+context$_{6}$ glyphs$_{12}$ graph$_{5}$ heat map$_{1}$ hierarchy$_{18}$ isosurface$_{4}$ magic lens$_{4}$ node-link$_{3}$ not applicable$_{5}$ novel visual encoding$_{34}$ parallel coordinates$_{54}$ pixel-based$_{9}$ progressive update$_{3}$ radviz$_{4}$ rendering enhancement$_{4}$ scatterplot$_{97}$ scatterplot pipeline stage:view transformation$_{1}$ skeleton$_{2}$ star coordinates$_{2}$ starplot$_{2}$ surfaces$_{8}$ treemap$_{4}$ volume visualization$_{5}$

other: clutter reduction$_{15}$ filtering$_{1}$ histogram$_{1}$ machine learning$_{6}$ matching$_{1}$

1. AlbuquerqueEisemannLehmann2010 [inproceedings] (2010)
Improving the visual analysis of high-dimensional datasets using quality measures
Albuquerque, Georgia Eisemann, Martin Lehmann, Dirk J Theisel, Holger Magnor, Marcus
Abstract: *Modern visualization methods are needed to cope with very high-dimensional data. Efficient visual analytical techniques are required to extract the information content in these data. The large number of possible projections for each method, which usually grow quadratically or even exponentially w...* ▶
pipeline stage:view transformation **user involvement:**computation centric **paper type:**technical **data type:**high-dimensional points **analysis method:**quality measure **visual method:**pixel-based **visual method:**radviz view optimization

2. AlbuquerqueLoweMagnor2011
Visualization and Computer Graphics, IEEE Transactions on (2011)
Synthetic Generation of High-Dimensional Datasets
Albuquerque, G. Lowe, T. Magnor, M.
Abstract: *Generation of synthetic datasets is a common practice in many research areas. Such data is often generated to meet specific needs or certain conditions that may not be easily found in the original, real data. The nature of the data varies according to the application area and includes text, graph...* ▶
pipeline stage:data transformation **user involvement:**computation centric **paper type:**technical **data type:**high-dimensional points **analysis method:**projection **visual method:**scatterplot

3. AnandWilkinsonDang2012 [inproceedings] (2012)
Visual pattern discovery using random projections
Anand, Anushka Wilkinson, Leland Dang, Tuan Nhon
Abstract: *An essential element of exploratory data analysis is the use of revealing low-dimensional projections of high-dimensional data. analysis method:Projection Pursuit has been an effective method for finding interesting low-dimensional projections of multidimensional spaces by optimizing a score func...* ▶
pipeline stage:data transformation **user involvement:**interactive exploration **paper type:**technical **data type:**high-dimensional points **analysis method:**projection **analysis method:**subspace **visual method:**parallel coordinates **visual method:**scatterplot

215 publications

Figure 3.1: Online library for the high-dimensional data visualization survey.

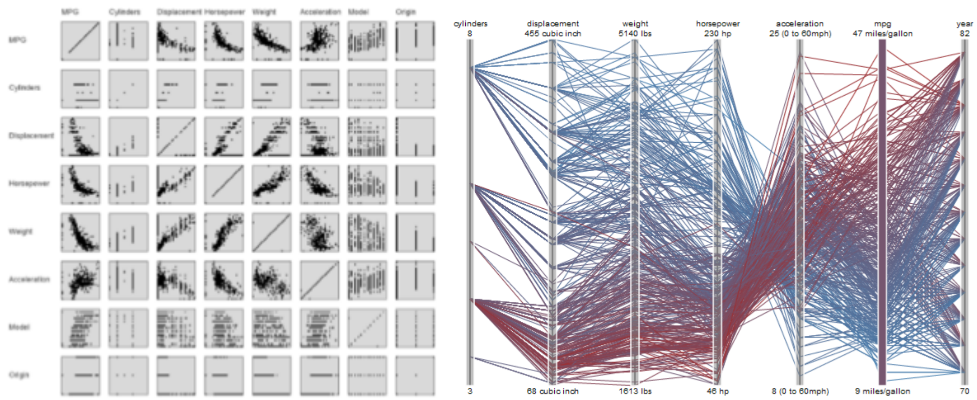


Figure 3.2: Scatterplot Matrix and Parallel Coordinate Plots.

primary drawbacks of SPLOMs is the scalability. The number of bivariate scatterplots increases quadratically with respect to the dataset's dimensionality. Numerous studies have introduced methods for improving the scalability of SPLOMs by automatically or semiautomatically identifying more interesting plots.

Originally introduced by John W. Tukey, Scagnostics are a set of measures designed for identifying interesting plots in a SPLOM. The recent works of Wilkinson et al. [1, 18] extend the concept to include nine measures (illustrated in Fig .3.3) capturing properties such as outliers, shape, trend, and density. In addition, they improve the computational efficiency by using graph-theoretic measures. Scagnostics have also been extended to handle time series data [34]. Guo [35] introduces an interactive feature selection method for finding interesting plots by evaluating the maximum conditional entropy of all possible axis-parallel scatterplots. The rank-by-feature framework [12, 20] allows users to choose a ranking criterion, such as histogram distribution properties and correlation coefficients between axes, for scatterplots in SPLOMs.

Data class labels can play an important role in identifying interesting plots and selecting a meaningful ranking order. Sips et al. utilize class consistency [36] as a quality metric for 2D scatterplots. The class consistency measure is defined by the distance to the center of the class or entropies of the spatial distributions of classes. Tatu et al. [19] introduce different metrics for ranking the "interestingness" of scatterplots and PCPs for both classified and unclassified datasets. For data with labels, a class density measure and a histogram density measure are adopted as ranking functions for the scatterplots.

The ranking order provides only an indirect way to assess the scatterplots. Lehmann et al. [37] introduce a system for visually exploring all the plots as a whole. By reordering the rows and columns in the SPLOMs, this method groups relevant plots in the spatial vicinity

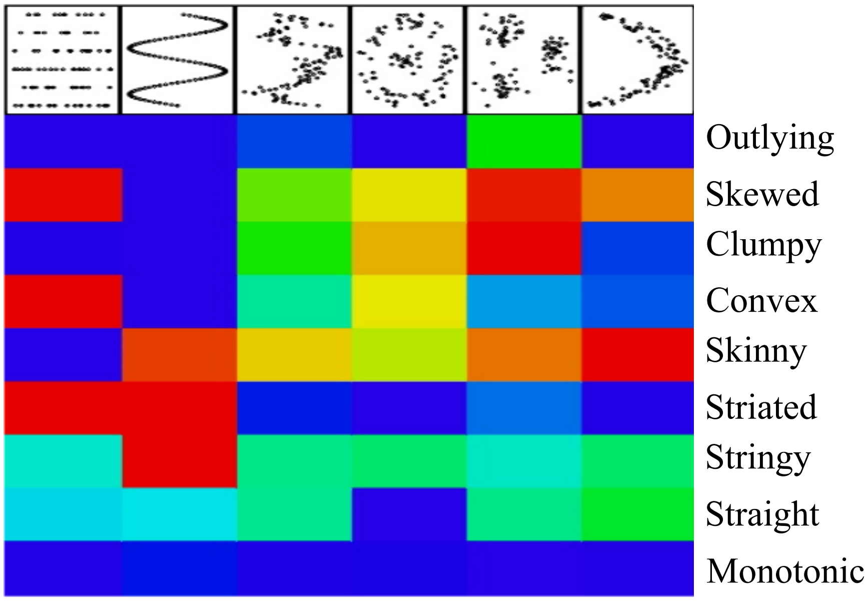


Figure 3.3: Graph-Theoretic Scagnostics introduced by Wilkinson et al. [1]

of one another. In addition, an abstraction can be obtained from the reordered SPLOM to provide a global view.

3.2 Parallel Coordinates

Compared to a SPLOM, for which only bivariate relationships can be directly expressed, the parallel coordinate plot (PCP) [38, 39, 40] allows patterns that highlight multivariate relations to be revealed by showing all the axes at once. For a given n-dimensional dataset, theoretically, there are $n!$ permutations of the ordering of the axes. With different axes order, vastly different information may be presented. Therefore, one of the fundamental challenges when dealing with PCPs is determining the appropriate orders of the axes [40]. Since a user typically can only interpret the visual patterns among nearby axes, the search space can be drastically reduced by focusing on localized axes orders, such as consecutive dimension triples (an axes and its immediate neighbors) or pairwise dimensions. For these scenarios, finding the minimum number of permutations needed to display all dimension triples or pairwise dimension combinations is the goal. Hurley et al. [41] adopt Eulerian tours and Hamiltonian decompositions of complete graphs to generate axis order permutations ($O(n/2)$) covering all bivariate patterns between dimensions. Inselberg has posed the problem of finding permutations that display all adjacent triples [39], which may be considered as a visualization challenge in PCPs.

A few other methods utilize quality metrics and subspace finding methods to automatically identify interesting axes orders. The PCP ranking methods developed by Tatu et al. [19] work for both classified and unclassified datasets. For unlabeled data, the Hough space measure is used, and for labeled data, a similarity measure and overlap measures are adopted. Ferdosi et al. introduce a dimension ordering method [42] that is applicable for both PCPs and SPLOMs utilizing the subspace analysis method from their earlier work [21] discussed in Section 3.4. Johansson and Johansson [43] propose an interactive system adopting a weighted combination of quality metrics for dimension selection and automatic ordering of the axes to enhance visual patterns such as clustering and correlation.

In addition, as the number of data points increases, the line density in the PCP increases dramatically, which can lead to visual clutter [40] thus hindering the discovery of patterns (e.g., density variation, dimension correlation). As a result, clutter reduction through filtering, aggregation, visual encoding, and dimension reordering, is another important challenge for PCPs. Interactive filtering of data, such as brushing linked axes, is essential for alleviating visual clutter. Chapter 10 of Inselberg's book [39] provides a great discussion on how to exploit interactivity in PCPs to understand large and complex data. A set of query operations, which can be combined to construct more complex queries, is identified as the basis for the exploration.

Aggregation and visual encoding can also be used in combination with interactive exploration to reduce visual clutter. In the work by Novotny and Hauser [44], a focus+context visualization scheme is adopted for reducing the clutter by aggregation. In this approach, the outliers are indicated by single lines and the trends that capture the overall relationship between axes are approximated by polygon strips. Zhou et al. introduce a line bundling scheme [45] for enhancing the visual clusters. The authors exploit the curved edges and arrange the edges by minimizing the curvature while maximizing the parallelism of the adjacent ones. The progressive parallel coordinate (PPC) [46] work introduces several LOD-hierarchy based visual encoding approaches to address the challenges of large datasets and overplotting. In the work introduced by Dang et al. [47], density is expressed by stacking overlapping elements. For the PCP case, a 3D visualization is presented, where either the edges are stacked as curves or the points on the axes are stacked vertically as dots to alleviate the clutter with an additional dimension. Finally, as dimension ordering can greatly affect the PCPs' expressiveness, Peng et al. [48] introduce a clutter reduction method for PCPs by reordering the axes.

3.3 Dimension Reduction

One of the fundamental techniques for analyzing high-dimensional datasets is dimension reduction. Dimension reduction techniques can be roughly divided into two major classes:

linear projection and manifold learning. The projection methods try to approximate the high-dimensional space through a linear subspace of lower dimensionality. If the data lie within such a space, they can be re-expressed by a linear basis transformation without loss of information. However, if the data are non-linear and lie on a manifold of lower dimensionality, then the linear subspace may not be able to capture the structure of the data faithfully. Instead, the distance relationships along this manifold can be learned in an unsupervised manner and generate a non-linear data mapping. These techniques are abstracted from Euclidean distance relationships and capture distances along a manifold. easily expressed as a matrix multiplication. Therefore, out of sample points can be projected into the same space without any additional effort. However, for manifold learning methods, examining the relationship between existing embedding and the out-of-sample points is a challenging task. Secondly, each axis in the linear projection results is a linear combination of the original dimensions. The linear relationship allows interpretation of the projection results. On the contrary, the manifold learning results are extremely difficult to interpret in respect to the original dimensions. Finally, the manifold learning methods are usually more computationally expensive compared to their linear counterparts, such as PCA (principal component analysis). Dimension reduction techniques are key components for many visualization tasks. Existing work either extends the state-of-the-art techniques, or improves upon their capabilities with additional visual aid.

Linear Projection. Linear projection uses linear transformation to project the data from a high-dimensional space to a low-dimensional one. It includes many classical methods, such as Principal component analysis (PCA), Multidimensional scaling (MDS), Linear discriminate analysis (LDA), and various factor analysis methods.

PCA [49] is designed to find an orthogonal linear transformation that maximizes the variance of the resulting embedding. PCA can be calculated by an eigendecomposition of the data's covariance matrix or a singular value decomposition of the data matrix. The interactive PCA (iPCA) [50] introduces a system that visualizes the results of PCA using multiple coordinated views. The system allows synchronized exploration and manipulations among the original data space, the eigenspace, and the projected space, which aids the user in understanding both the PCA process and the dataset. When visualizing labeled data, class separation is usually desired. Methods such as LDA aim to provide a linear projection that maximizes the class separation. The recent work by Koren et al. [51] generalizes PCA and LDA by providing a family of flexible linear projections to cope with different kinds of data.

Non-linear Dimension Reduction. There are two distinct groups of techniques in non-linear dimension reduction, under either the metric or non-metric setting. The graph-based techniques are designed to handle *metric* inputs, such as Isomap [52], Local Linear Embedding (LLE) [53], and Laplacian Eigenmap (LE) [54], where a neighborhood graph is used to capture local distance proximities and to build a data-driven model of the space.

The other group of techniques address non-metric problems commonly referred to as non-metric MDS or stress-based MDS by capturing non-metric dissimilarities. The fundamental idea behind the non-metric MDS is to minimize the mapping error directly through iterative optimizations. The well-known Shepard-Kruskal algorithm [55] begins by finding a monotonic transformation that maps the non-metric dissimilarities to the metric distances, which preserves the rank-order of dissimilarity. Then, the resulting embedding is iteratively improved based on stress. The progressive and iterative nature of these methods has been exploited recently by Williams et al. [56], where the user is presented with a coarse approximation from partial data. The refinement is on-demand based on user inputs. Others rely on hybrid methods [57, 58] based upon stochastic sampling and interpolation to approximate the solution. t-SNE [59] has gained a lot of attention recently due to its effectiveness for visualizing high dimensional data in 2D. t-SNE utilizes a probability distribution to encode the inter-point neighborhood information, and a mismatched probability distribution is used between high- and low-dimensional spaces to eliminate the unwanted attractive forces, therefore, resolving the crowding problem [59].

Control Points Based Projection. For handling large and complex datasets, the traditional linear or non-linear dimension reductions are limited by their computational efficiency. Some recent developments (e.g., [60, 61, 62, 63, 64]), utilize a two-phase approach, where the control points (anchor points) are projected first, followed by the projection of the rest of the points based on the control points location and local features preservation. The general paradigm is illustrated in Figure 3.4. Such designs lead to a much more scalable system. Furthermore, the control points allow the user to easily manipulate and modify the outcome of the dimension reduction computation to achieve desirable results.

Distance Metric. For a given dimension reduction algorithm, a suitable distance metric is essential for the computation outcome as it is more likely to reveal important structural information. Brown et al. [27] introduce the distance function learning concept, where a new distance metric is calculated from the manipulation of point layouts by an expert user. In [65], the author attempts to associate a linear basis with a certain meaningful concept constructed based on user-defined examples. Machine learning techniques can then be employed to find a set of simple linear bases that achieve an accurate projection according

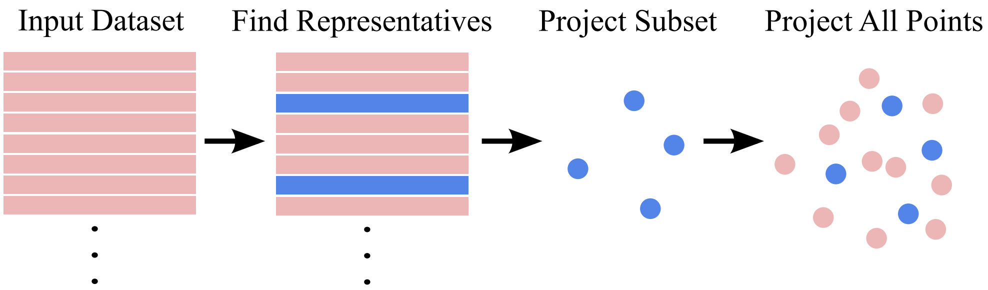


Figure 3.4: Control points based projection. The representative points (control points) are selected for initial projection, and the subsequent projection of all the dataset is accelerated by utilizing the information from the initial projection.

to the prior examples. The structure-based analysis method [66] introduces a data-driven distance metric inspired by the perceptual processes of identifying distance relationships in parallel coordinates using polylines.

Dimension Reduction Precision Measure. One of the fundamental challenges in dimension reduction is assessing and measuring the quality of the resulting embeddings. Lee et al. introduce the ranking-based metric [67] that assesses the ranking discrepancy before and after applying dimension reduction. This technique is then generalized [68] and used for visualizing dimension reduction quality. A projection precision measure is introduced in [69], where a local precision score is calculated for each point with a certain neighborhood size. In the distortion-guided exploration work [31], several distortion measures are proposed for different dimension reduction techniques, where these measures aid in understanding the cause of highly distorted areas during interactive manipulation and exploration. For MDS, the stress can be used as a precision measure. Seifert et al. [70] further develop this idea by incorporating the analysis and visualization for better understanding of the localized stress phenomena.

3.4 Subspace Clustering

Dimension reduction aims to compute one single embedding that best describes the structure of the data. However, this could become ineffective due to the increasing complexity of the data. Alternatively, one could perform subspace clustering, where multiple embeddings can be generated through clustering either the dimensions or the data points, for capturing various aspects of the data.

Dimension Space Exploration. Guided by the user, dimension space exploration methods interactively group relevant dimensions into subsets. Such exploration allows us to better understand their relationships and to identify shared patterns among the dimensions. Turkay et al. introduce a *dual* visual analysis model [4] where both the dimension embedding and point embedding can be explored simultaneously. Their later improvement [71] allows for the grouping of a collection of dimensions as a *factor*, which permits effective exploration of the heterogeneous relationships among them. The Projection Matrix/Tree work [72] extends a similar concept to allow a recursive exploration of both the dimension space and data space. Several visual encoding methods also rely on the concept of dimension space exploration.

Subsets of Dimensions. Compared to the dimension space exploration, where the user is responsible for identifying patterns and relationships, subspace clustering/finding methods automatically group related dimensions for identifying clusters in these subspaces. Subspace clustering filters out the interferences introduced by irrelevant dimensions, allowing lower-dimensional structures to be discovered. These methods, such as ENCLUS [13], originate from the data mining and knowledge discovery community. They introduce some very interesting exploration strategies for high-dimensional datasets, and can be particularly effective when the dimensions are not tightly coupled. The $TripAdvisor^{ND}$ [16] system employs a sightseeing metaphor for high-dimensional space navigation and exploration. It utilizes subspace clustering to identify the sights for the exploration. The subspace search and visualization work [15] utilizes the SURFING (subspaces relevant for clustering) [14] algorithm to search the high-dimensional space and automatically identifies a large candidate set of interesting subspaces. For the work presented by Ferdosi et al. [21], morphological operators are applied on the density field generated from the (3D) PCA projection of the high-dimensional data for identifying subspace clusters.

Non-Axis-Aligned Subspaces. Instead of grouping the dimensions, which essentially creates axis-aligned linear subspaces, identifying non-axis-aligned subspaces is a more flexible alternative. Projection Pursuit [9] is one of the earliest works aimed at automatically identifying the interesting non-axis-aligned subspaces, where the projections are considered to be more interesting when they deviate more from a normal distribution. Recently, some advances have been made in the machine learning community to perform non-axis-aligned subspace clustering [22]. Instead of finding (possibly overlapping) clusters in axis-aligned subspaces defined by different dimensions combinations, the points are directly clustered together for sharing similar linear subspaces. In particular, this approach assumes the complex structure of the data can be approximated by a mixture of linear subspaces (of varying dimensions), and each of the linear subspaces corresponds to a set of points where their relationships can be approximately captured by the same linear subspace. Lehmann et al. [73] have recently introduced an interesting and different approach for identifying a set of distinct linear projections. By adopting a dissimilarity measure, they aim to remove duplicated data patterns by optimizing the dissimilarity among the selected projections. By utilizing random projection [74], Anand et al. [75] introduce an efficient subspace finding algorithm for data with thousands of dimensions. The algorithm generates a set of candidate subspaces through random projections and presents the top-scoring subspaces in an exploration tool.

3.5 Topological Data Analysis

A crucial step in gaining insights from large, complex, high-dimensional data involves feature abstraction, extraction, and evaluation in the spatiotemporal domain for effective exploration and visualization. Topological data analysis (TDA), a new field of study (see [76, 77, 78, 79, 80, 81] for seminal works and surveys), has provided efficient and reliable feature-driven analysis and visualization capabilities. Specifically, the construction of topological structures [82, 83] from scalar functions on point clouds (e.g., Morse-Smale complexes, contour trees, and Reeb graphs) as "summaries" over data is at the core of such TDA methods. Reeb graphs/contour trees capture very different structural information of a real-valued function compared to the Morse-Smale complexes as the former is contour-based and the latter is gradient-based (Figure 3.5). They both provide meaningful abstractions of the high-dimensional data, which reduces the amount of data needed to be processed or stored; and they utilize sophisticated hierarchical representations capturing features at multiple scales, which enables progressive simplifications of features differentiating small and large scale structures in the data.

Morse-Smale Complexes. The Morse-Smale complex (MSC) [84, 85] describes the topology of a function by clustering the points in the domain into regions of monotonic

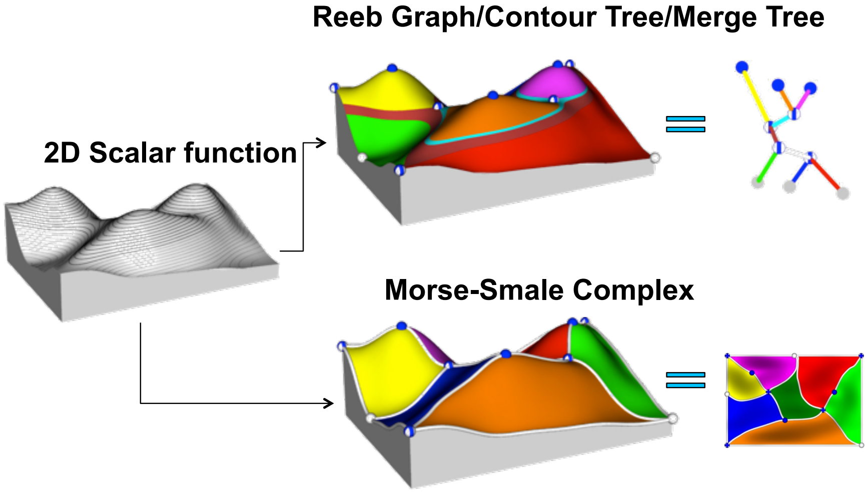


Figure 3.5: Contour- and gradient-based topological structure of a 2D scalar function.

gradient flow, where each region is associated with a sink-source pair defined by local minima and maxima of the function. The MSC can be represented using a graph where the vertices are critical points and the edges are the boundaries of areas of similar gradient behavior. The simplification of the MSC is obtained by removing pairs of vertices in the graph and updating connectivities among their neighboring vertices, merging nearby clusters by redirecting the gradient flow. MSCs have been shown to be effective in identifying, ordering, and selectively removing features of large-scale data in scientific visualizations (e.g., [86, 87, 88]).

HDViz [8] employs an approximation of the MSC (in high dimensions) to analyze scalar functions on point cloud data. It creates a hierarchical segmentation of the data by clustering points based on their monotonic flow behavior, and designs new visual metaphors based on such a segmentation. Correa et al. [89] suggest that by considering a different type of neighborhood structure, we can improve the accuracy in the extracted topology compared to those obtained within HDViz.

Reeb Graphs and Contour Trees. The Reeb graph of a real-valued function describes the connectivity of its level sets. A contour tree is a special case of Reeb graph that arises in simply-connected domains. The Reeb graph stores information regarding the number of components at any function value as well as how these components split and merge as the function value changes. Such an abstraction offers a global summary of the topology of the level sets and enables the development of compact and effective methods for modeling and visualizing scientific data, especially in high dimensions (i.e., [90, 91]). Mapper [91] decomposes data into a simplicial complex resembling a generalized Reeb graph, and visualizes the data using a graph structure with varying node sizes. The work has developed into a startup company AYASDI. Their software is shown to extract salient features in a study of diabetes by correctly classifying normal patients and patients with two causes of diabetes [92] and various other applications [93, 94].

Efficient algorithms for computing the contour tree [95] and Reeb graph [96] in arbitrary dimensions have been developed. A generalization of the contour tree has been introduced by Carr et al. [97, 98] called the joint contour net (JCN), which allows for the analysis of multi-field data.

Other Topological Features. Ghrist [81] and Carlsson [80] both offer several applications of TDA and in particular highlight the topological theory used in a study of statistics of natural images [99]. Wang et al. [100] utilize TDA techniques developed by Silva et al. [101] to recover important structures in high-dimensional data containing non-trivial topology. Specifically, they are interested in high-dimensional branching and circular structures. The circle-valued coordinate functions are constructed to represent such features. Subsequently, they perform dimension reduction on the data while ensuring such structures are visually preserved.

3.6 Model Manipulation

User interactivity is an integral part of many high-dimensional data visualization techniques. Based on the amount of user interaction, we can classify all high-dimensional data visualization methods into three categories: computation-centric, interactive exploration, and model manipulation. The distinction between interactive exploration and model manipulation is made to emphasize a particular manipulation paradigm, where the underlying data model is modified based on interaction to reflect user intention. The difference among these paradigms are illustrated in Figure 3.6. Computation-centric approaches require only limited user input such as setting initial parameters. Interactive exploration approaches navigate, query, and filter the existing model interactively for more effective visual communication. Model manipulation techniques represent a class of methods that integrate user manipulation as part of the algorithm, and update the underlying model to reflect the user input to obtain new insights.

Take the distance function learning work [27] for example. The initial embedding is created using a default distance measure. Through interaction, the initial point layout is modified based on the expert user's domain knowledge. The system then adjusts the underlying distance model to reflect the user input. Hu et al. present a method [102] for improving the translation of user interaction to algorithm input (visual to parameter interaction) for distance learning scenarios. The explainers [65] are projection functions created from a set of user-defined annotations. The control point based projection methods [60, 61, 62, 63, 64] update the overall projection result based on user manipulation of the control points. In the iLAMP method [103], inverse projection extrapolation is used for generating synthetic multidimensional data out of existing projections for parameter space exploration. In the Local Clustering Operation work [104], the visual structure is modified in PCPs through user-guided deformation operators. Finally, Liu et al. [31] allow for direct manipulation of the dimension reduction embedding to resolve structural ambiguities. The interactively updated distortion measure is used for feedback during manipulation.

3.7 Animation Enhancement

As stated in Heer et al.'s work [105], animation, when used appropriately, can significantly improve graphical perception. Many techniques for visualizing high-dimensional

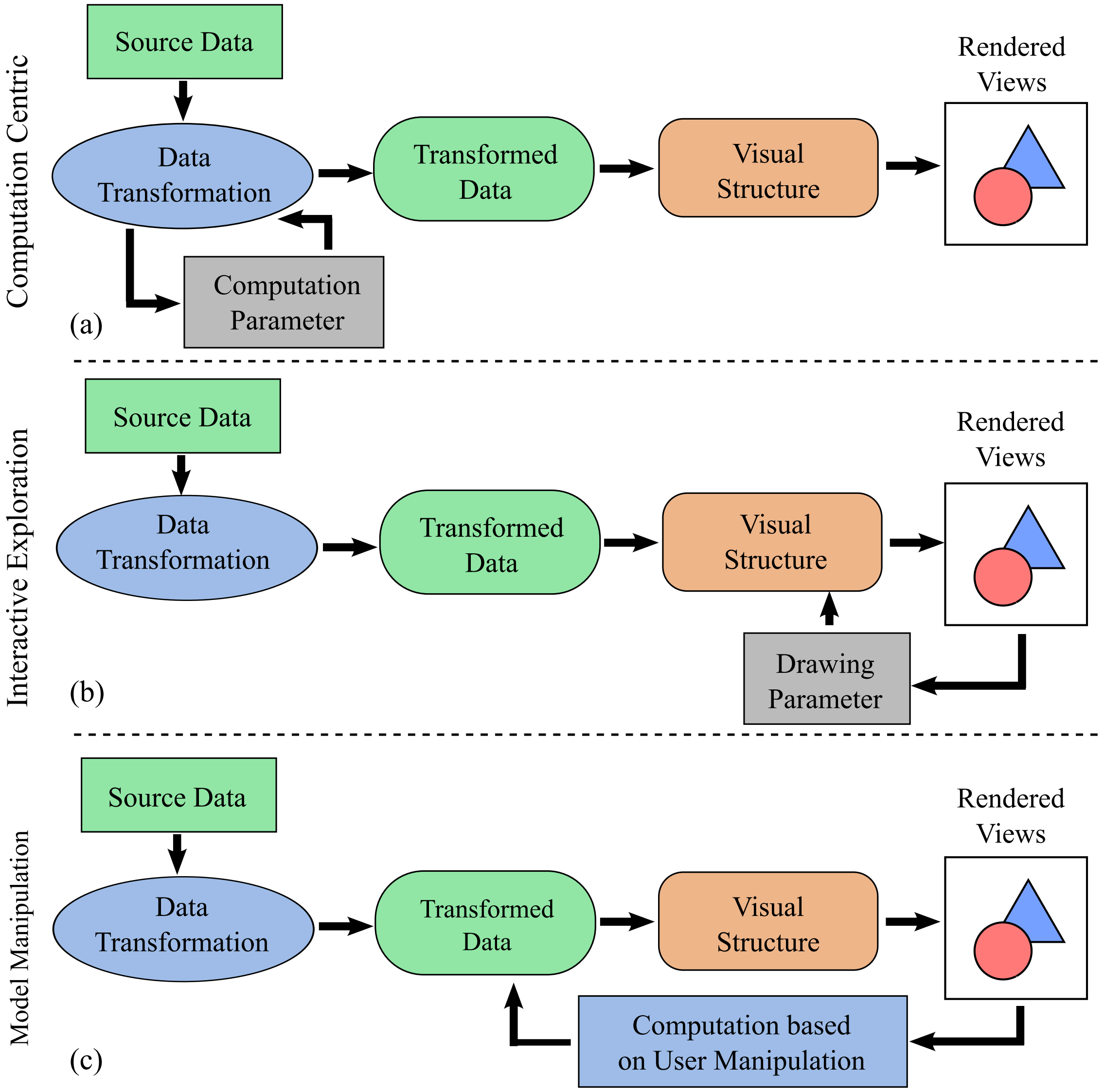


Figure 3.6: The different user interaction paradigms: computation-centric, interactive exploration, model manipulation.

data utilize animated transitions to enhance the perception of point and structure correspondences among multiple relevant plots (views of the data).

The GGobi system [106] provides a mechanism for calculating a continuous linear projection transition between any pair of linear projections based on the principal angles between them. In the Rolling the Dice work [107], a transition between any pair of scatterplots in a SPLOM is made possible by connecting a series of 3D transitions between scatterplots that share an axis. RnavGraph [108] constructs a graph connecting a number of interesting scatterplots. A smooth animation is generated between all scatterplots that are connected by an edge. The $TripAdvisor^{ND}$ [16] system allows users to explore the neighborhood of a subspace by tilting the projection plane using a polygonal touchpad interface.

PART II

SUBSPACE ANALYSIS FOR IDENTIFY INFORMATIVE 2D PROJECTIONS

CHAPTER 4

SUBSPACE ANALYSIS AND DYNAMIC PROJECTION

One of the fundamental challenges of visualizing high-dimensional data through 2D projections is how to reduce the enormous exploration space, so that human users are not overwhelmed by the number of visual representations. To strike a balance between capturing the intrinsic structures and generating interpretable results, 2D linear projections are selected as the focus of this research. For a given dataset, there are infinite ways a 2D linear projection can be generated; how to identify the informative ones that capture the intrinsic structure and aid in the understanding of the data is the goal of this study.

The fundamental idea behind this work (illustrated in Figure 1.5) is that high-dimensional data can be decomposed into multiple subsets, each part of which is representable in a lower-dimensional space. Such a strategy provides a "divide and conquer" approach for addressing the complexity of high-dimensional space. In this research, the subspace clustering algorithm [22, 24] that originated in the machine learning community is adopted for capturing the low-dimensional subset of the data. Once the data are clustered into subspaces based on their intrinsic low-dimensional structures, the linear basis that supports each subspace naturally defines a number of interesting 2D projections (views), without the need to rank their interestingness explicitly [109, 1]. On the other hand, when there are outliers or the subspaces intersect, subspace clusters may not be perfect. To estimate the dimension and basis of each subspace, applying traditional dimension estimation (e.g., PCA) to the subspace clusters may produce suboptimal results (see Section 4.1.2). In this research, a novel dimension and basis estimation algorithm tailored for visualization is introduced for identifying 2D projections in these subspaces. Compared to PCA (Section 4.1.2), this algorithm is less susceptible to outliers or intersecting subspaces, and can better discriminate the different subspaces. The combined effort of subspace clustering and the novel dimension and basis estimation algorithm is referred to as *subspace analysis*.

Despite each 2D projection providing valuable localized information, without understanding their relationships the user may still not be able to obtain global insights regarding the data. To best utilize these informative 2D projections identified via subspace analysis, a navigation graph is constructed in this research to visualize the distance between 2D linear projections. In addition, animated transitions (dynamic projection [110]) are provided among these projections to aid in the understanding of their relationships, especially in terms of point correspondence and structure similarity.

The combination of subspace analysis with dynamic projection transition also addresses some visualization challenges in dynamic projection. Since the promotion of exploratory data analysis by John W. Tukey, a few methods have been introduced that utilize the dynamic projection to aid in the understanding of the high-dimensional datasets. Grand tour [111] generates a continuous projection (i.e., a tour) that attempts to cover the entire high-dimensional space. Even though the use of animated transitions is proven to be effective in conveying structural information, the complexity of the high-dimensional space requires a lengthy tour that prevents effective exploration. A more recent work [110] tries to address such issues by making projection pursuit results [112] the targets along the tour's path. However, the projection pursuit is optimized for the entire space, which may fail to capture even very simple linear structures in the subsets of the data. In addition, organizing data analysis as a sequential tour limits the user's involvement in the exploratory process. In this research, the issues that potentially prevent effective use of dynamic projections are addressed by utilizing subspace analysis to identify a set of projections that capture the intrinsic structure of the data. By introducing a navigation graph that aids flexible navigation among these projections, intuitive exploration of the high-dimensional space is achieved.

As illustrated in Figure 4.1, the proposed framework [28] contains two major components: subspace analysis and interactive exploration. The subspace analysis (highlighted in the blue box) is responsible for subspace identification and basis estimation. The visual exploration (highlighted in the orange box) enables users to visualize and interact with the subspace analysis results. It generates subspace views (2D projections marked by colored rectangular boxes) from the corresponding basis, creates the navigation infrastructure (the view navigation graph), and produces animated transitions between 2D projections generated from subspaces (the subspace views). The transition from the black subspace view to the yellow 2D subspace view is illustrated in the figure. The interactive exploration communicates with the subspace analysis when a clustering or model estimation parameter is modified, triggering a recomputation of the subspace information.

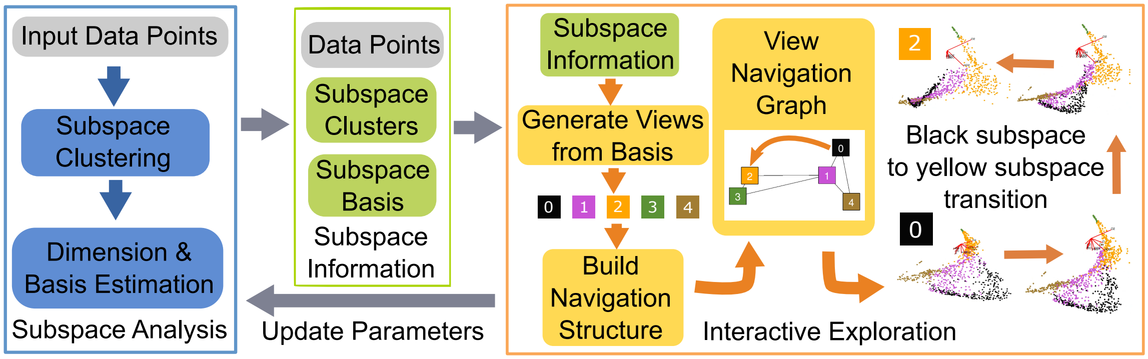


Figure 4.1: An overview of the visualization workflow.

4.1 Subspace Analysis

The underlying assumption of fitting a single linear subspace makes PCA ineffective in modeling complex, high-dimensional data. In this work, a more general assumption of fitting a union of subspaces is considered. An existing subspace clustering approach is adopted to partition data into multiple subspaces. For visualization purpose, 2D linear projections need to be generated from these subspaces. As a result, a novel technique is proposed to estimate the parameters of each subspace (dimension and basis).

4.1.1 Subspace Clustering

Let us assume that the set of samples $\{\mathbf{x}_i \in \mathbb{R}^D\}_{i=1}^T$ is drawn from an unknown union of $n \geq 1$ linear subspaces $\{S_j\}_{j=1}^n$. The dimensions of the subspaces, $0 < d_j < D$ $(j = 1, \cdots, n)$, are unknown and each subspace is described as $S_j = \{\mathbf{x} \in \mathbb{R}^D : \mathbf{x} = \mathbf{U}_j\mathbf{y}\}$, where $\mathbf{U}_j \in \mathbb{R}^{D \times d_j}$ is a basis for the subspace S_j and $\mathbf{y} \in \mathbb{R}^{d_j}$ is the low-dimensional representation of a sample $\mathbf{x}$. When $n = 1$, this problem reduces to PCA. A wide variety of algorithms have been proposed in the machine learning literature to determine the multiple subspaces [22], and in particular methods based on spectral clustering [113] have been very effective.

Spectral clustering requires an *affinity* matrix $\mathbf{A} \in \mathbb{R}^{T \times T}$, where A_{ij} measures the similarity between samples i and j [113]. Subspace clustering is a special case where $\mathbf{A}$ captures the subspace relationships, i.e., samples belonging to the same subspace have a strong affinity between them. In particular, the affinity matrix is constructed by representing each sample as a linear combination of other samples, i.e., $\mathbf{X} \approx \mathbf{XW}$, s.t. $W_{ii} = 0$ $(i = 1 \cdots T)$. Here, $\mathbf{W} = [\mathbf{w}_i]_{i=1}^T$ is the affinity matrix and the condition $W_{ii} = 0$ ensures that a sample is not used for its own reconstruction. Since this problem is highly *ill-posed*, different forms of *regularization* (e.g., sparsity, low-rank) can be considered [24]. In addition to allowing the user to specify the number of clusters, I also integrate the spectral clustering auto-tuning method [114] to aid the selection. To provide some intuition, a simple synthetic dataset is used to help illustrate the process. The dataset contains two intersecting 2D planes embedded in 3D. As shown in Figure 4.2 the subspace clustering identifies two subspace clusters that correspond to the two planes, respectively.

4.1.2 Subspace Construction

Basis Estimation. Given the subspace associations, using PCA on samples belonging to each cluster can provide the basis spanning that subspace. However, since PCA attempts to determine directions of maximal variance, outliers that might arise due to subspace clustering can significantly affect this process. Instead, a more general graph embedding approach is proposed that allows the exploitation of the relationships between the different subspaces (encoded in the affinity matrix) to discriminate the different subspaces and improve the resilience to outliers.

The affinity matrix constructed during subspace clustering will contain strong edges between samples within a subspace and weak edges across subspaces. A block-diagonal matrix is extracted from the affinity matrix $\mathbf{W}$, corresponding to only the samples in that subspace to compute the basis vectors. For a subspace S_j, the set of indices of samples belonging to the respective cluster is denoted by Λ_j. I solve the following optimization problem to estimate the basis:

$$\mathbf{U}_j = arg\min_{\mathbf{U}} \sum_{i \in \Lambda_j} \left\| \mathbf{U}^T \mathbf{x}_i - \sum_{k \neq i, k \in \Lambda_j} W_{ik} \mathbf{U}^T \mathbf{x}_k \right\|_2^2 \quad \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}.$$

Here the matrix $\mathbf{I}$ is the identity matrix, $\mathbf{U}_j \in \mathbb{R}^{D \times d_j}$ contains the set of basis functions, and d_j is the dimension of the subspace. The solution to this problem can be obtained using generalized eigenvalue decomposition.

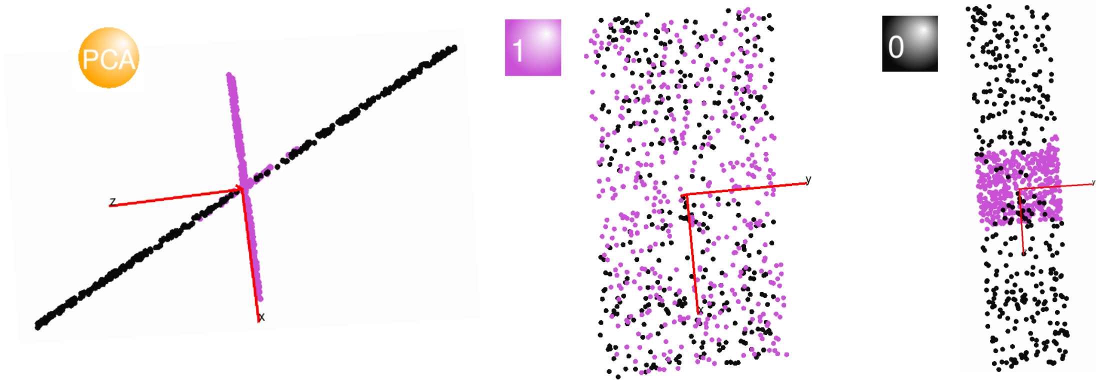


Figure 4.2: An intuitive explanation of the subspace clustering. Left: The PCA view shows the projection from the side of the two 2D planes. By subspace clustering, two 2D subspaces are obtained (middle and right), each corresponds to a plane.

Dimension Estimation. The basis estimation process assumes the knowledge of the subspace dimension, d_j. The proposed dimension estimation technique relies on the assumption that the basis set estimated for a cluster must be ineffective in describing samples from other clusters. This is achieved by picking the dimension that results in the maximal separation of a subspace from the subspace estimated using all samples not belonging to the cluster considered. For a subspace S_j, the set of samples $\{\mathbf{x} \in S_j\}$ is used to estimate the basis $\mathbf{U}_j$, whereas the out-of-sample basis $\bar{\mathbf{U}}_j$ is obtained using the samples $\{\mathbf{x} \notin S_j\}$. The dimension d_j is varied between 2 and $D-1$, measure the distance between the $\mathbf{U}_j \in \mathbb{R}^{D \times d_j}$ and $\bar{\mathbf{U}}_j$ in each case, and pick the dimension where sufficient separation is achieved. The subspace separation is measured using the Grassmannian distance (see Section 2.3).

Comparison to PCA. By using the USPS digits dataset as an example, the superior performance of the proposed basis and dimension estimation approach is demonstrated. The USPS handwritten digits dataset contains 2500 images belonging to 10 classes [115]. Here three different analysis strategies are compared: (i) A global PCA subspace for the entire data, (ii) Estimate a PCA subspace for each class independently, and (iii) Estimate a subspace for each class using the proposed approach. In the first case, all samples are projected onto the single PCA subspace and with a fixed neighborhood size ($k = 10$), for each sample, I measure the number of samples in the neighborhood that share its class label. For cases (ii) and (iii), I measure the neighborhood recovery performance for each class by projecting all samples onto its corresponding subspace. Figure 4.3 shows the average accuracy for each of the classes, obtained using the three approaches, along with their corresponding subspace dimensions.

As expected, the single linear PCA subspace is insufficient for describing the complex relationships in the dataset and has the least accuracy in all cases. Even with the union of subspace assumption, using PCA to estimate the basis can erroneously project samples from different classes close to each other and hence its performance is only marginally better. Finally, by considering the relationships between the different subspaces, the proposed method faithfully recovers the neighborhood.

4.2 Visual Exploration of the Subspaces

Through the subspace analysis, a simplified representation of the high-dimensional space in the form of low-dimensional linear subspaces is acquired. For each subspace, a set of 2D views (projections) can be generated in a similar fashion as the scatterplot matrix, i.e., by choosing all pairs of vectors from the basis. To better understand these views and their relationships, I organize them in a multi-level View Navigation Graph. The exploration of

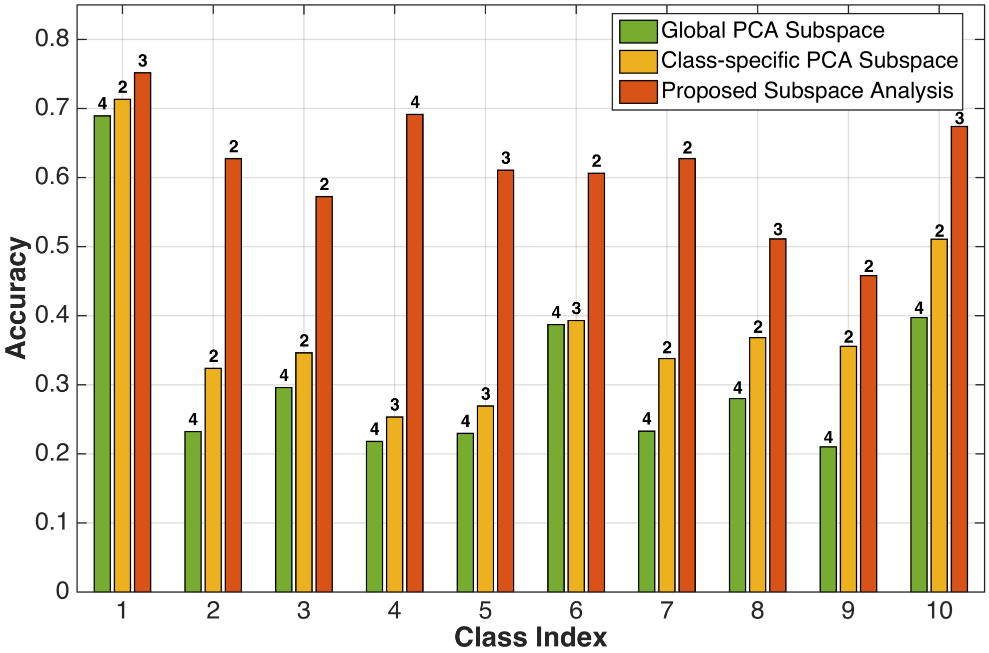


Figure 4.3: For the subspace corresponding to each class, the average accuracy of samples in finding neighbors sharing their class label is shown, using different subspace analysis strategies. The subspace dimension is also showed in each case.

the subspaces focuses on the manipulation of the graph and the seamless transitions between individual 2D projections (views). However, a direct linear interpolation between the point locations leads to non-linear and uninterpretable frames in the animation. In the proposed framework, the dynamic projection approach [116, 117] is adopted, where the animation is defined by a set of intermediate linear subspaces that smoothly transition from one 2D subspace to another. The pipeline of the interactive exploration is illustrated in Figure 4.1.

4.2.1 The Grassmann Distance

Understanding the distance between subspaces or 2D projections is crucial for the exploration. As discussed in Section 2.3, by introducing the concept of Grassmannian, distance between subspaces can be defined easily. Given two subspaces on a Grassmannian manifold, represented by their orthonormal bases, $\mathbf{A}$ and $\mathbf{B}$ of size $D \times d$, the distance measured along the geodesic is the *Grassmann distance*. The geodesic distance can be computed by decomposing $\mathbf{A}^T\mathbf{B}$ using its SVD and obtaining $\sum_{i=1}^{d} \left(\theta_i^2\right)^{\frac{1}{2}}$. Here, θ_i denotes a principal angle and is obtained as $\cos^{-1} \sigma_i$, where σ_i is the corresponding singular value. When considering two subspaces of different dimensions [32], $\mathbf{A} \in Gr(d_1, D)$ and $\mathbf{B} \in Gr(d_2, D)$ (with $d_1 < d_2$), the distance can be calculated by finding a d_2-dimensional plane $\mathbf{C}$ contained in $\mathbf{B}$ that is closest to $\mathbf{A}$, and measuring the distance between $\mathbf{A}$ and $\mathbf{C}$. Given two projections, the intermediate subspaces created through dynamic projection [116, 117] are points along the shortest geodesic path between the two. Importantly, each frame in the animation is indeed a linear projection. Hence, comparing two subspaces is equivalent to comparing their basis sets that span the subspaces. Note that the commonly adopted Euclidean distance is not an appropriate metric for comparing subspace basis sets. Linear subspaces are known to lie on a Grassmannian manifold, and hence the geodesic distance on this manifold allows accurate comparison of subspaces. In contrast to to existing subspace comparison approaches [16], the Grassmann distance is invariant to the ordering of the basis vectors and axis rotations within a subspace (for example, the rotation of the 2D projection orientation within the 2D plane). Estimating the Grassmann distance involves SVD evaluation, making it computationally more expensive. Hence, in this work I resort to using a computationally efficient distance metric on the Grassmannian, the Chordal distance [32].

4.2.2 View Navigation Graph

The subspace views (i.e., 2D linear projections), defined by all pairs of vectors in the basis, are generated for each subspace. Compared to the scatterplot matrix or other subspace clustering methods that try to find axis-aligned features, the proposed technique produces a much smaller number of views. However, without proper organization, navigating among these views can still be daunting. The *view navigation graph* (Figure 4.4) is introduced to help manage the views and guide the exploration. Instead of displaying all the views together, I organize the views into groups corresponding to their respective subspaces. Each group (a subspace) has a representative view (i.e., projection), defined by the two most dominant basis directions.

The user can start the initial exploration with only the representatives of each subspace. In the view navigation graph (Figure 4.4), each subspace representative is denoted by a square glyph marked with the subspace dimension at its lower right corner. All the representative nodes are connected via a k-nearest neighborhood (kNN) graph constructed from the Grassmann distance between subspaces. Such a graph provides a global overview of the subspaces and captures the inter-subspace relationships. The user can then expand each three or higher dimensional subspace for a more focused study. During the expansion, the selected representative is replaced by a subgraph formed by all individual 2D views generated from the subspace basis. Such a dynamic graph construction ensures interactive, multi-scale exploration of the space of subspace views. Although the choice of k can be important for the kNN graph, Figure 4.5 demonstrates that regarding the usage in this

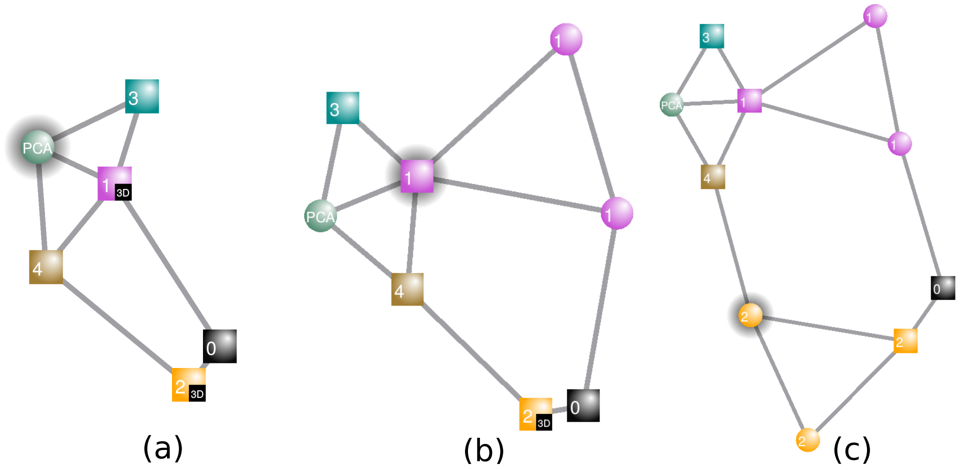


Figure 4.4: The views navigation graph. (a) The square glyph indexed by subspace ID corresponds to the representative view of a given subspace. The circle glyph corresponds to a non-representative view or the PCA projection. For each subspace with dimension three or higher, we can dynamically *expand* its representative into multiple 2D views generated from its basis (e.g., (b) & (c)).

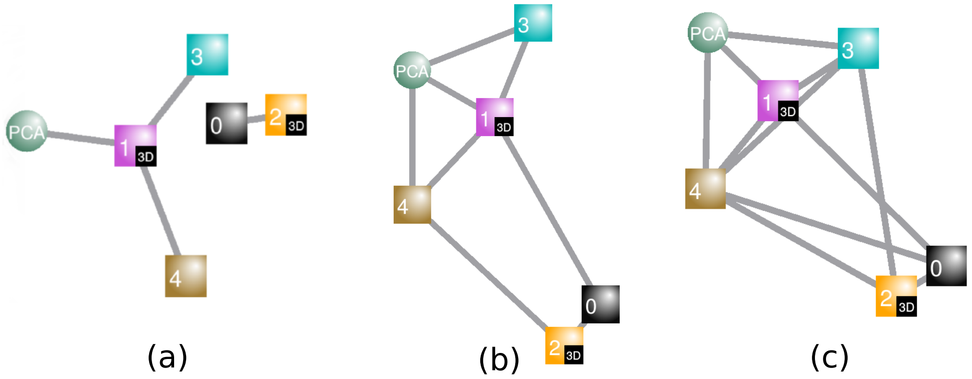


Figure 4.5: kNN graphs with varying k. (a) $k = 1$. (b) $k = 2$. (c) $k = 3$. From all of the graphs (a)-(c), we can infer two groups of subspaces with strong intra-cluster relationships: the orange and black subspaces; and the PCA, brown, purple, and cyan subspaces.

research, a small variation in the choice of k does not have a great impact on understanding the inter-cluster relationships. Other alternative neighborhood graphs can be considered for future study, such as the Gabriel graph [118] or β-skeletons [119]. It would be interesting to define these graphs beyond the Euclidean metrics, that is, in the setting of Grassmann distance.

4.3 System Implementation

Software Architecture. As part of the *DataExplorerHD* software framework, the proposed system architecture (Figure 4.6) is designed to be easily configurable and extentable.

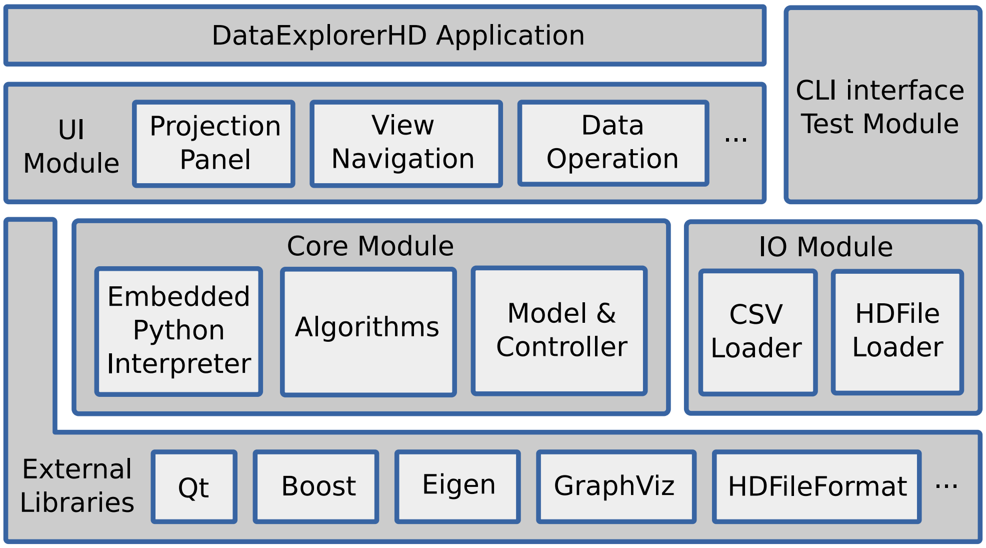
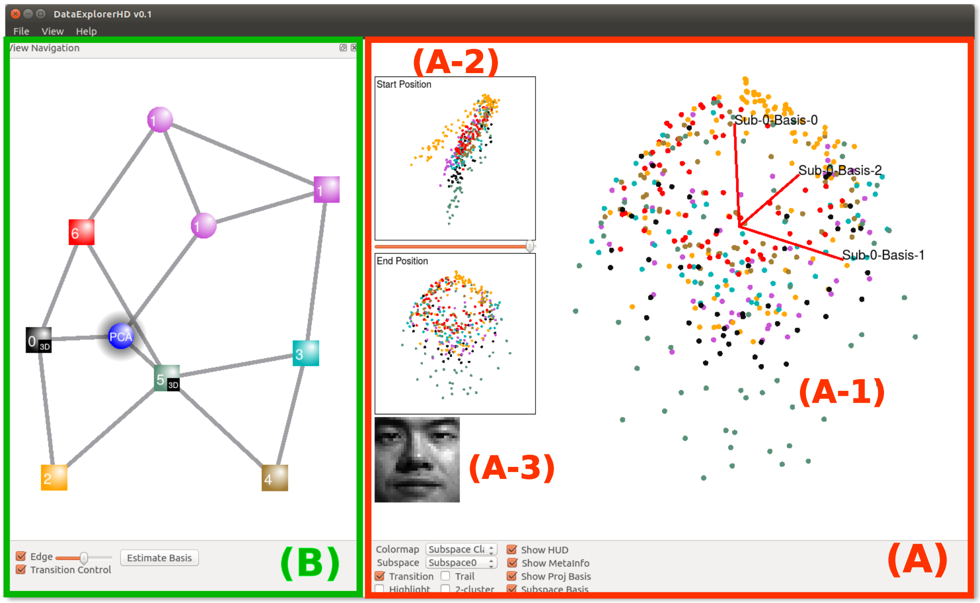


Figure 4.6: The software architecture.

It provides infrastructures for combining different components to create an environment adaptable for future demands.

The core functionalities are implemented in C++, and Qt is used for all the GUI and drawing tasks. The architecture consists of several major modules. The *Core module* includes the essential algorithms and abstract data models and operations. The *IO module* handles all the tasks related to the file IO. I design an XML-based binary file format and its accompanying library, where new types of data can be easily integrated. The *UI module* includes individual GUI components (view navigation graph panel, dynamic projection panel, parallel coordinates, data operation panel, etc.), which can be customized for different tasks. To provide the utmost flexibility, the tool integrates an embedded Python interpreter in the *Core Module*, which enables the seamless integration of Python script and C++ code. Such a design allows us to implement the subspace clustering code in Python, taking advantages of fast prototyping, quick iterations, and readily available machine learning libraries. Since the Python implementation contains mostly matrix computation, which indirectly invokes the C library, the speed of the implementation is comparable to an optimized C/C++ implementation (the performance and scalability issues are discussed in Section 7.6).

User Interface and Interaction. Figure 4.7 shows the interface of the system when it is configured for interactive exploration tasks. (A) is the main display panel demonstrating the dynamic projections (A-1) at its center. Each projection is augmented with a bi-plot (which consists of axes that correspond to basis vectors scaled by their coefficients). Alongside the projection view (A-1), two small insets are included: (A-2) shows both the source and the



☑ Edge
☑ Transition Control
Estimate Basis

Colormap: Subspace Cl
Subspace: Subspace0
☑ Transition ☐ Trail
☐ Highlight ☐ 2-cluster
☑ Show HUD
☑ Show MetaInfo
☑ Show Proj Basis
☑ Subspace Basis

Figure 4.7: User interface. (A) The dynamic projection panel that includes (A-1) the linear projection display, (A-2) the source and target views of the current animated transition with slider. (A-3) links to meta-information (e.g. images) of the original data if applicable. (B) The subspace view navigation panel.

target projections, where the slider between the thumbnails allows the user to play the animation back and forth; (A-3) presents the meta-information of the data (e.g., images) when available. (B) is the view navigation panel that contains the view navigation graph, which provides an interface for guiding the exploration process.

4.4 Application Examples

4.4.1 Combustion Simulation Dataset.

This dataset contains a collection of 2.8K samples from a large-scale combustion simulation [120]. Each sample is drawn from a 10D input parameter space that corresponds to the concentrations of 10 chemical compounds (e.g., H_2, O_2) involved in the simulation, with the temperature as the observed variable (the spatial information is not modeled here as the focus is the parameter space of the chemical concentrations). Scientists are interested in understanding how input parameters affect the local minimum temperature observed under the extinction and re-ignition phenomenon.

As shown in the view navigation graph (Figure 4.8(a)), the subspace analysis of this dataset gives three 2D subspaces (#0-black, #4-brown, and #3-cyan) and two 3D subspaces (#1-purple and #2-orange). The subspace views belong to two well-separated clusters in the view navigation graph: The cyan, purple, and brown subspace views are positioned in

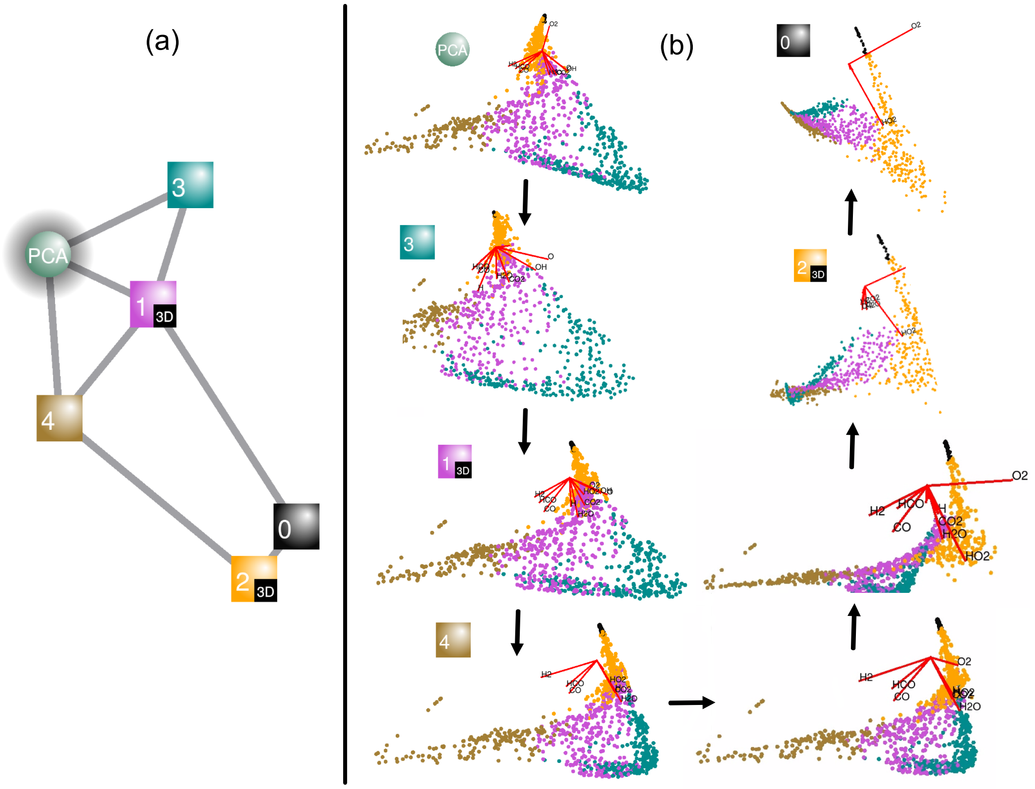


Figure 4.8: Combustion dataset. (a) View navigation graph. (b) From left to right, top to bottom: transition from the PCA view, to the cyan, purple, and brown subspace views; then to the orange, and finally to the black subspace view. Two snapshots of the dynamic transition between the orange and the black subspace views connected by black arrows are included.

proximity to each other; and similarly for the black and orange subspace views. A PCA view is also added to the view navigation graph.

Via dynamic projections, exploration is started from the PCA view to the cyan, purple, and brown subspace views sequentially, as illustrated in Figure 4.8(b). These views are close to one another in the view navigation graph. A small amount of tilting is observed during such transitions, indicating small rotational angles among basis vectors of these subspaces. Such observation likely indicates that these three subspaces are approximations of a gently curved, non-linear structure in the data. When transitioning from the brown subspace view to the orange one, a drastic expansion of the orange cluster and a compression of the brown, purple, and cyan clusters are observed. This animation indicates that the orientation of the orange subspace is very different from the previous three subspaces. Finally, the transition from the orange to the black subspace view demonstrates their similarities in terms of the small rotational angle. These observations give user an intuitive understanding of the structure in the data, namely, the cyan, purple, and brown subspaces share structural similarities; the orange and black subspaces are closely related; yet both sets of subspaces are structurally very different.

Further insights regarding the data could be obtained by close examination of the dynamic transitions between the PCA view and the orange subspace view. The PCA finds the best single linear subspace to represent the data but fails to capture the structure of each subspace with equal accuracy.

As shown in Figure 4.9(a), relatively high inaccuracy is observed in the circled region (that corresponds to the orange subspace) based on projection distortion measures [31, 68]. This is due to the fact that PCA maximizes variance across all dimensions while the orange subspace contains only two dominant dimensions (i.e., O_2 and HO_2 in its bi-plot in Figure 4.9(c)) with large variance.

On the other hand, when transitioning from the PCA view to the orange subspace view, intrinsic structure of the orange subspace is better preserved while the high distortion region is shifted elsewhere (Figure 4.9(b)). In addition, through the orange subspace view, additional understanding of the extinction pheonomina is obtained. As highlighted in Figure 4.9(c)-(d), temperature profile (c) indicates two distinct local minima (pointed by two red arrows) in the data, while the HO_2 concentrations (d) exhibit significant variations surrounding these minima (pointed by two red arrows). According to the domain experts, the differences in the HO_2 concentration correspond to two distinct types of extinction conditions, one of which is not readily visible in the PCA view.

4.4.2 Yale Face Dataset.

The Yale face dataset is a subsample from the original database[17]. It consists of 439 face images from seven people, which can be roughly labeled as (in no particular order): one African female, one Asian female, two Asian males, one Caucasian male, one Indian male, and one Middle Eastern male. During the visual analysis, I assume the true labels

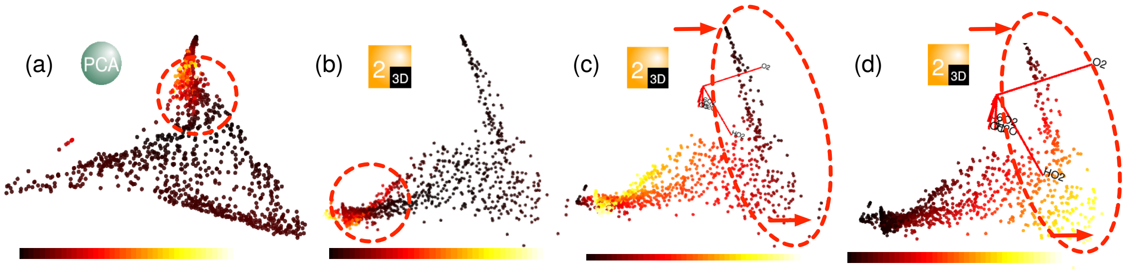


Figure 4.9: Combustion dataset. (a) PCA view colored by pointwise distortion measure. (b) Yellow subspace view colored by pointwise distortion measure. (c) Yellow subspace view colored by temperature. (d) Yellow subspace view colored by HO_2 concentration.

are unknown at the moment and later use these labels to validate the observations. The original images have a resolution of 32 × 32. Random projection is used to reduce their resolution to 10 × 10; therefore, the points are embedded in 100D space. As shown in the view navigation graph (Figure4.10(a)), the subspace analysis gives four 2D subspaces (#2-orange, #3-cyan, #4-brown, #6-red) and three 3D subspaces (#0-black, #1-purple, #5-green). Let's start the exploration of the data from the PCA view (Figure 4.10(b)). Although the PCA view gives poor separations among different subspace clusters, points from each cluster are arranged in a circular fashion according to the continuously varying lighting directions. This observation helps us examine the shifts in lighting conditions within target subspace views during dynamic projections.

Now let's transition from the PCA view to the orange subspace view (Figure 4.11(a)). A rotational motion around a horizontal axis is observed, and the transition end with a side angle view of the data. In the orange subspace view (Figure 4.11(b)), the green, purple, and orange clusters form three stratified sets. By validating with the face images, these three clusters contains mostly images from an Asian female and two Asian males, respectively. Furthermore, the *amount* of shadow in the images increases along the dominating direction of each cluster towards its overlapping region. In addition, as illustrated in Figure 4.11(c), the misclassified points (highlighted in the dotted circle) appear at the top of the embedding that correspond to the face images where most facial features are in deep shadows. Similarly, when transitioning from the PCA view to the brown and cyan subspace views, respectively, clear class separations among the target subspace views can be observed. That is, the brown and the cyan clusters (mostly contains images of an Indian male and a Caucasian

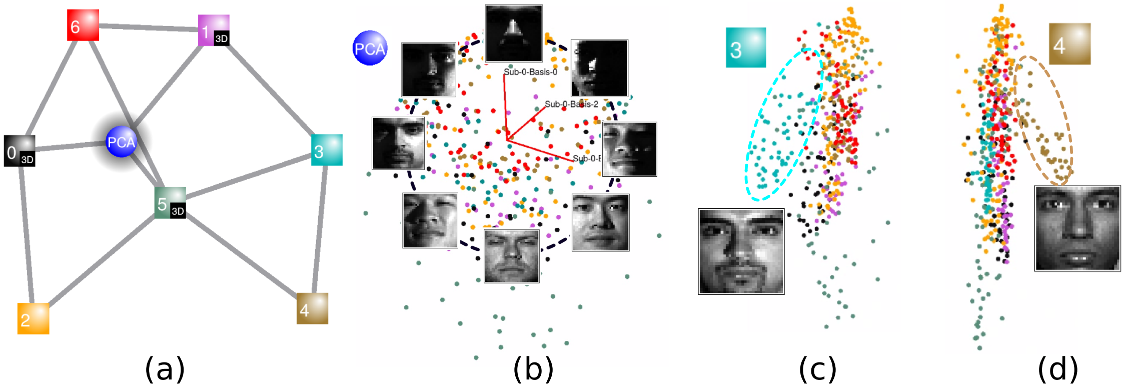


Figure 4.10: Yale face dataset. (a) View navigation graph. (b) illustrates the correlation between the points distribution and the lighting directions in the PCA view. (c) The cyan subspace view. (d) The brown subspace view.

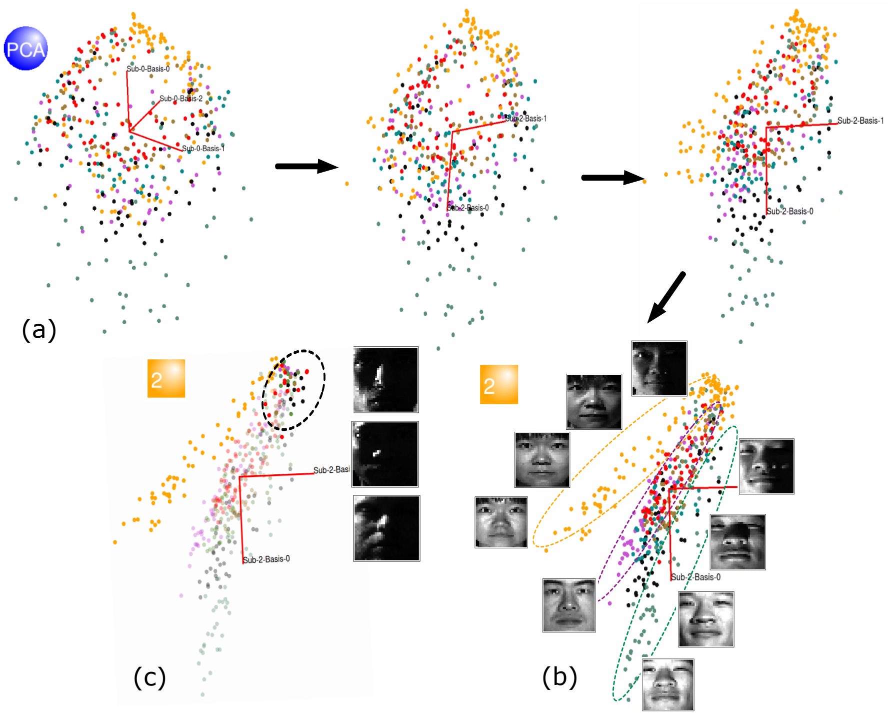


Figure 4.11: Yale face dataset. (a)-(b) Dynamic transition from the PCA to the orange subspace view; two snapshots of the animations are included. (b) Shows the three stratified sets and highlights the image variation (the amount of shadow) along their dominant directions. (c) highlights the mis-classification (circled area) caused by poor lighting conditions.

male, respectively) are shown to be well-separated from the rest of the data points (see Figure 4.10(c)-(d)).

Finally, when transitioning from the PCA view to the red subspace view (which contains mostly images of an African woman), a slightly different rotation is observed. The resulting embedding does not exhibit clear class separation between the red cluster and the remaining points (Figure 4.12 (a)-(b)). Further exploration (Figure 4.12 (c)) reveals that along the dominant direction, the images in the red cluster vary according to the *directions of lighting*. This trend is very different from that the one green, purple, and orange subspace clusters (which all contain images of people of Asian origin) share, where images vary along the dominating direction according to the *amount of shadow*. Such a distinction between the two groups is likely caused by the differences in facial features and skin tone.

4.4.3 MNIST Dataset

The MNIST dataset is sampled from the MNIST handwritten digits database. The original images have a resolution of 28×28. I down-sample the images into 12×12 and use

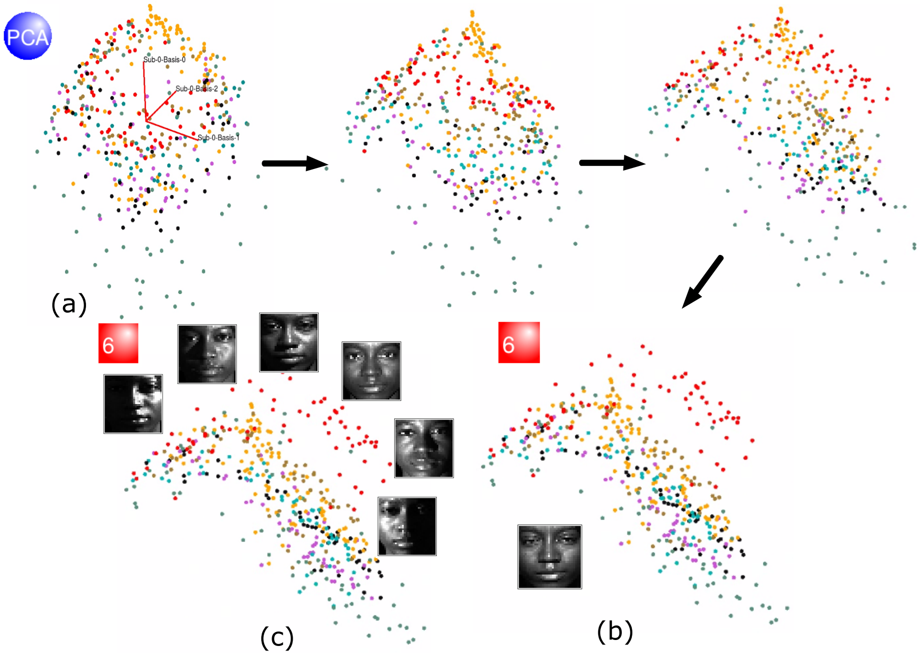


Figure 4.12: Yale face dataset. (a)-(b) Dynamic transition from the PCA to the red subspace view; two snapshots of the animation are included. (c) Shows the red points in the red space view where their corresponding images vary along the cluster's dominating direction according to the differences in lighting direction.

500 samples for better interactive performance (in terms of generating smooth animations). Four 3D subspaces (#1-purple, #5-green, #8-dark red, #9-pink) and six 2D subspaces (#0-black, #2-orange, #3-cyan, #4-brow, #6-red, #7-blue) is obtained from the subspace analysis. During the visual analysis, the true labels are assumed unknown and later are used to validate the observations.

As demonstrated in the combustion dataset, the compression and expansion types of motions during the dynamic projection likely indicate substantial structural differences between source and target subspaces. Here I give a few more examples. When applying dynamic transitions between the black and the purple subspaces (Figure 4.13), the black cluster drastically expands while the purple clusters compresses into a very small cluster. Such motions is illustrated using the motion trails.

Referring back to the original handwritten images, it turns out that the purple cluster contains the handwritten digit “1” while the black subspace contains the digit “0”. Therefore it is not surprising that these two subspaces appear to lie on the opposite sides of the high-dimensional space. Further observation of the black subspace view indicates that points in the black subspace are distributed according to the width of the digits, that is, points on the left correspond to “fat” handwritings while the ones on the right are “skinny”. When

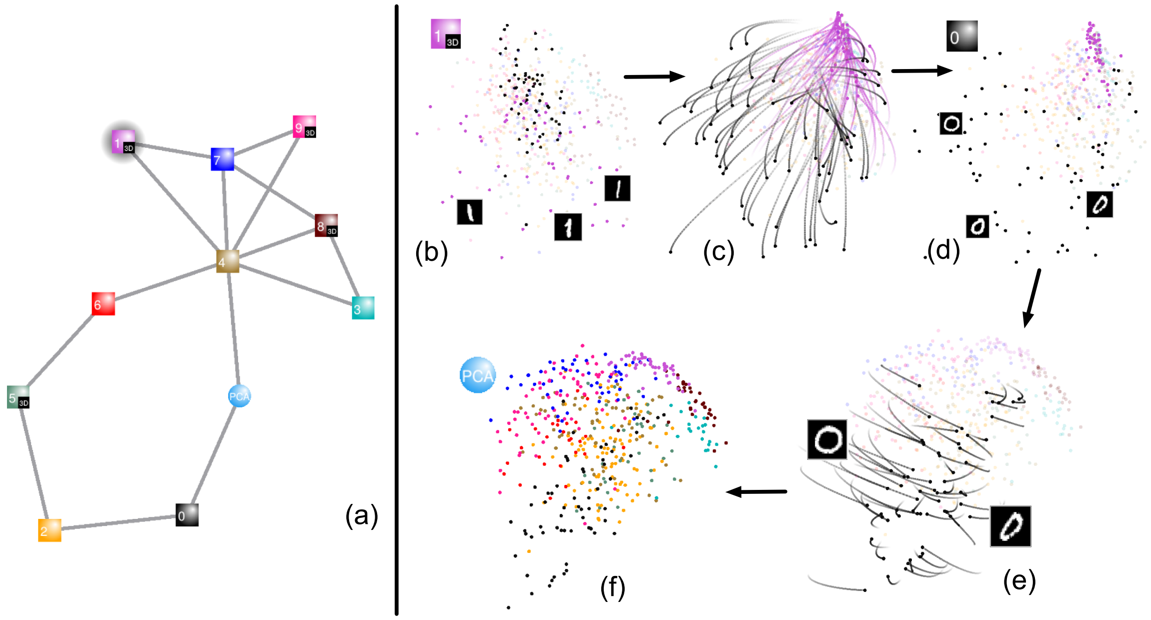


Figure 4.13: MNIST. (a) View navigation graph. Dynamic projections from the purple (b) to the black (d) subspace view, and then to the PCA view (f). Motion trails (c & e) are used to highlight such transitions in the static images.

continuing the transition from the black subspace view to the PCA view, black points from both sides of the projection moves towards a central vertical line, therefore the obtained PCA embedding no longer preserves local structure (i.e., distribution of digits according to their widths) compared to the black subspace view. Similar behavior exists during the dynamic transition between the red and the blue subspace views (Figure 4.14). Referring back to the original images, the distinct shapes of “6” (mostly red) and “7” (mostly blue) contribute to the very different subspaces these points define.

In addition, the images of the digit “1” is split into three overlapping clusters (cyan, dark red, and purple) in the PCA view (Figure 4.15). They correspond to the different

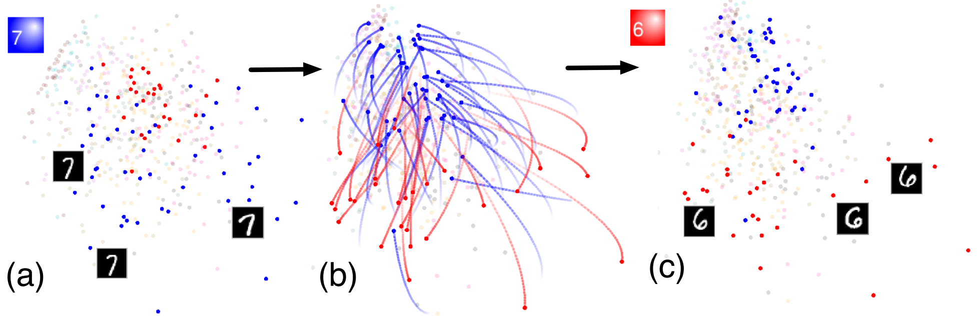


Figure 4.14: MNIST. Dynamic projection from the red (a) to the blue (c) subspace view. A motion trail (b) is used to highlight the transitions in the static image.

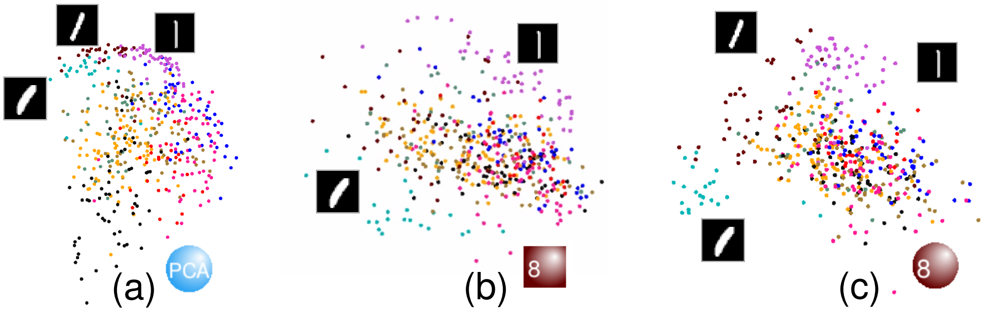


Figure 4.15: MNIST. (a) PCA view, where the difference in orientation of the digit "1" is captured by the overlapping cyan, dark red and purple cluster. (b)-(c) Two of the three red subspace views where digits "1" become separated based on their orientations.

orientations of the digit. When transitioning to two of the dark red subspace views, the tilted digits "1" and the straight ones become well-separated onto the opposite sides of the embeddings. This example again demonstrates the subspace's ability to preserve local features. The transitions from the PCA to the subspace views help the user study the relationships between the global and local structures.

4.5 Evaluation and Discussion

4.5.1 Comparisons with Existing Systems.

In order to better understand the difference between the proposed framework and existing techniques, a comparison among three relevant systems: GGobi [116], Scatterplot dice [107], and $TripAdvisor^{ND}$ [16] is presented below. These systems either utilize some forms of subspace-finding algorithms or use animated transitions between a pair of 2D views for data exploration.

The *GGobi* [116] system utilizes the dynamic projection by defining a series of transition target projections, either by random generation [111] or by switching among different projection pursuit indices (e.g., *holes*, *central mass*). Due to the random nature of such transitions, it may take significantly longer time for a user to identify the informative views representing meaningful structures. Meanwhile, the projection pursuit indices try to capture a pre-defined set of properties, which may not be meaningful for a given dataset. Such limitation is illustrated in Figure 4.16, in which the GGobi system is applied to the example datasets based on the *holes* index (a few frames are captured within the dynamic projection results). A "hole"-like structure is detected by the projection pursuit index within the face dataset, but such a structure does not exist for the combustion dataset. In the proposed framework, the source and target views are obtained through subspace analysis, which naturally captures the intrinsic structure of the data. With the help of the

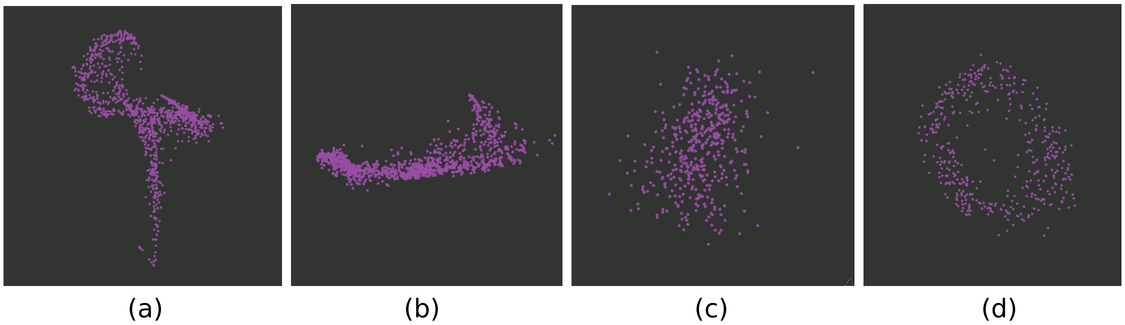


Figure 4.16: GGobi results using the grand tour and projection pursuit *holes* index; example frames for the combustion (a)-(b) and face datasets (c)-(d).

view navigation graph that captures the relationships among individual projections, the proposed system provides a more structured approach in exploring the space of projections and revealing important structures.

The *Scatterplot dice* [107] approach is built on top of the scatterplot matrix. A 3D transition between a pair of plots in the scatterplot matrix can be obtained when they share one axis (i.e., shared dimension). The system automatically generates a series of 3D animations to connect any two plots. The system is easy to understand, and the animations provide valuable information. However, one of the fundamental limitations of such a system is the lack of scalability as the number of dimensions goes up. One of the examples in this work contains a 100D dataset. Using the scatterplot matrix, the user will end up with a large number of unique projections that are almost impossible to be explored interactively.

The $TripAdvisor^{ND}$ [16] system provides a Focus+Context approach, where a number of "tourist sites", each corresponding to the best view of each subspace (the subset of dimensions), is given as an overview of the data. The user can delve into each of these tourist sites for a more focused study by tilting the projection plane around a local neighborhood. The proposed framework differs from the $TripAdvisor^{ND}$ in three ways. First, instead of finding related subsets of dimensions, the proposed approach decomposes the data into clusters, each represented by a simple (not necessarily axis-aligned) linear subspace. Second, compared to an ad-hoc similarity measure, a distance measure between a pair of views is defined rigorously through the *Grassmann distance*. Third, while $TripAdvisor^{ND}$ allows local neighborhood exploration around one projection, the proposed framework allows full transitions among multiple structural-revealing projections, and helps the user obtain insights via both local and global exploration.

4.5.2 Interviews with the Experts.

To better evaluate the usability of the proposed tool, and in particular, the effectiveness of dynamic projections, I conduct in-depth interviews with two computer science faculties, one in machine learning (Expert A) and one in information visualization (Expert B). I obtain their opinions and suggestions on various aspects of the system.

Expert A finds the tool to be useful in "providing an alternative, interesting way to visualize high-dimensional data", compared to the traditional dimensionality reduction methods. The subspaces captured by the algorithm reveal local linear relationships that may otherwise be hidden by a projection optimized for global properties (such as PCA). Local views are linked by the navigation graph to form a global picture. To evaluate the effectiveness of dynamic projections, Expert A first inspects individual subspace views, and then he enables and explores animated transitions between them. He states that "the animated transition is very useful in tracking changes between two projections, and the transitions are easy to follow." In addition, each frame is computed from a linear projection, thereby making it easy to interpret the animation. Expert A also suggests I include other linear projections methods (e.g., Linear Discriminate Analysis for labeled data) in the tool to obtain additional insights. Since high-dimensional data visualization techniques are indispensable for better understanding machine learning algorithms, Expert A is interested in using the tool for visualizing certain natural language processing (NLP) word vector datasets; such a collaboration yields the work discussed in Chapter 6.

Expert B points out that the most significant advantage of using dynamic projection in the tool is the ability to track the correspondences among individual points between the starting and ending projections; such correspondences could be further highlighted by enabling motion trails (an optional visual component implemented in the current system). Combining the dynamic transitions with cluster labels, the user can infer the overall changes easily in cluster configurations. Expert B emphasizes that extra caution is needed when inferring high-dimensional structures based on the intuitions obtained from the 2D space. He suggests that a slider be added to allow the user to play the animated transitions back and forth, which could facilitate the understanding of dynamic projection. I have integrated this functionality in the tool.

4.5.3 System Scalability and Flexibility.

The usability of the tool depends greatly on its scalability and flexibility. The subspace clustering ($O(n^2k)$) and basis estimation ($O(k^2)$) algorithm have a combined time complexity of $O(n^2k+k^2)$ (where the n is the number of points, and k is the number of dimensions).

For the example datasets, the subspace analysis computation takes between 15-120 seconds on an Intel Core i5 2.8GHz desktop computer. The system allows both runtime turning of model parameters and pre-computation with multiple parameter configurations. The n^2 factor limits the subspace clustering algorithm for processing extremely large datasets directly. However, by utilizing smart sampling and summarization, I have been able to scale the system to handle very large datasets that contains several million points [29]. To handle a large data dimension (e.g., the face dataset), random projection can be applied to reduce the dimension to a manageable size. With a volume rendering extension, the core functionality of the system can be adopted for designing multi-dimensional transfer function for visualizing multivariate volume dataset [29] (discussed in Chapter 5), which exemplifies the flexibility of the proposed framework.

CHAPTER 5

DESIGN MULTIVARIATE TRANSFER FUNCTION VIA SUBSPACE ANALYSIS

Multivariate volumetric datasets arise naturally from many scientific applications, such as fluid dynamics, combustion, and climate simulations, where various physical measurements (e.g., temperature, pressure, and velocity) or multiple chemical species in the parameter space are used to define complex features in the volumetric space. With the explosive growth of such datasets, providing fast and effective tools for their analysis and visualization becomes increasingly challenging. For example, how do we interactively visualize large volumes, and how do we intuitively explore high-dimensional parameter space for volume visualization? The latter problem is particularly challenging, as it does not benefit substantially from powerful hardware and instead requires fundamental algorithmic advances.

One possible solution is to understand the interdependencies and joint effects of multiple variables and use this information to design a transfer function (TF) that links the parameter space to the volumetric space. Direct volume rendering with scalar volume data assigns colors and opacities to every voxel in the volume through a 1D TF. Kniss et al. [121] extend the TF design space from 1D to 3D by adding the gradient information. However, designing high-dimensional TFs is especially challenging, since no direct representation of high-dimensional space is possible. Some recent approaches utilize brushing operated on 2D embeddings of the data points in the parameter space obtained by dimension reduction [122] or graph drawing [123], with the assumption that such embeddings provide relatively good structural approximations to the data. In practice, many datasets often contain complex structures that are not easily unraveled by a single 2D embedding.

In this research, by utilizing the subspace analysis (introduced in the Chapter 4), multiple informative representations of the multivariate parameter space are captured. Instead of designing the transfer function in a single static canvas as in the traditional 2D transfer function, the subspace analysis and dynamic projections approach provides a flexible interface to explore the high-dimensional design space of the TF. The system [29] provides users with a variety of 2D projections that are designed to highlight intrinsic low-dimensional structures of the parameter space. One can infer relationships among these projections by exploiting the dynamic transitions between them, which provides an intuitive way for the user to explore the parameter space by creating a multifaceted, dynamic mental map of the data. A set of TF design paradigms specifically tailored to the dynamic design space is also introduced. The automatic TF design assigns colors based on subspace labels and provides a default visualization of the volumetric data. The semiautomatic TF design is guided by subspace labels and allows more refined TF design by exploiting relations among different parts of the data via animated transitions between viewing angles. The manual TF design treats subspace labels as latent information and exercises more freedom and flexibility in the design process. In order to scale the subspace analysis based approach to large volumetric datasets, intelligent subsampling and clustering strategies are employed to acquire a compact and meaningful representation of the data. The summarization of raw volume data greatly reduces the computation time, which enables interactivity during the TF design process. In summary, the proposed system not only provides a more intuitive understanding of the parameter space but also allows TF design on a dynamic canvas, thereby eliminating typical drawbacks of a single static 2D design space. A number of interactive techniques are also included as part of the TF design tools in the dynamic canvas, including dynamic subspace highlighting, multiple view sculpting, and neighborhood selection. By utilizing multivariate volumetric datasets from real-world applications, the effectiveness of the proposed approach is demonstrated.

5.1 Overview of the Computation Pipeline

Figure 5.1 gives an overview of the proposed visualization pipeline. The subspace analysis work discussed in Section 4 is used to explore the high-dimensional parameter space for visualizing multivariate volumetric data. Dynamic projections, together with subspace analysis, provide a multi-view canvas for effective TF design. Although I build upon infrastructure developed previously for high-dimensional data exploration, substantial extensions have been made to provide TF design and volume rendering capabilities, as well as address the scalability issue when handling large volumetric data.

The multivariate volume visualization interface consists of several interlinked panels, as illustrated in Figure 5.2. The dynamic projection panel (A) allows manipulation of 2D views of the data and enables TF design. Each subspace view in (A-1) is augmented with a biplot (i.e. where each attribute variable is displayed as a vector and the lengths

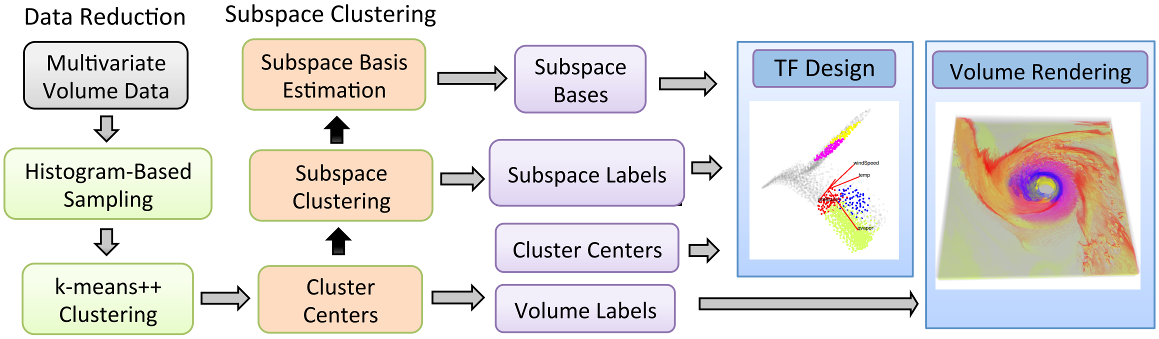


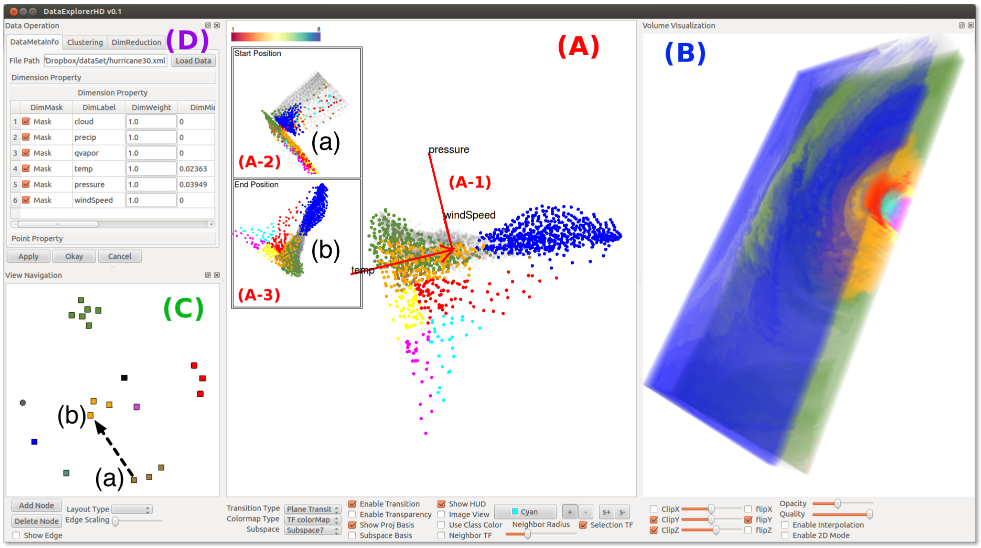
Figure 5.1: Overview of the visualization pipeline. There are three stages: 1) data reduction via sampling and k-means++ clustering; 2) subspace clustering of the reduced data; 3) TF design through parameter space exploration based on dynamic projections.

of the vector represents the coefficient with respect to the basis vector of the subspace) that provides a reference to the attribute dimensions. The volume visualization panel (B) displays the rendering result. The subspace navigation panel (C) allows the user to navigate among different views and control the animated transitions between them, where each node (draw as colored squares) represents a subspace view and views from the same subspace are grouped together with the same color (the only exception is the PCA view drawn as a colored circle). As showed in the figure, the nodes marked (a) and (b) correspond to the source and target views displayed in (A-2) and (A-3), respectively. The data panel (D) serves as the portal for data-centric operations such as applying dimension reduction and displaying meta information.

5.2 Data Reduction and Subspace Clustering

A volumetric dataset is typically large in scale compare to other dataset, but many of its data points may share similar feature vectors in the parameter space (e.g., data points in the empty space have feature vectors close to a constant). I exploit such feature redundancy to reduce the data size and preserve its structure through intelligent sampling and scalable clustering. Such data reduction is also necessary to ensure real-time interactivity and smooth animations during dynamic projections due to the memory and computation constraints.

I represent the volumetric dataset with a set of carefully selected points in the parameter space. First, a histogram of the data points is constructed in the parameter space based on their ℓ_2-norm, and sample a fixed percentage (e.g., 30%) of points within each discrete interval of the histogram to ensure good coverage of the parameter space. Second, I perform k-means++ clustering on the sampled points, obtain the cluster centers during convergence and approximate these cluster centers with their nearest neighbors within the



File Path /Dropbox/dataSet/hurricane30.xml

Dimension Property

	DimMask	DimLabel	DimWeight	DimMi
1	☑ Mask	cloud	1.0	0
2	☑ Mask	precip	1.0	0
3	☑ Mask	qvapor	1.0	0
4	☑ Mask	temp	1.0	0.02363
5	☑ Mask	pressure	1.0	0.03949
6	☑ Mask	windSpeed	1.0	0

Layout Type
Edge Scaling
☐ Show Edge

Transition Type Plane Transit
Colormap Type TF colorMap
Subspace Subspace7

☑ Enable Transition
☐ Enable Transparency
☑ Show Proj Basis
☐ Subspace Basis
☑ Show HUD
☐ Image View
☐ Use Class Color
☐ Neighbor TF
Neighbor Radius
☑ Selection TF

☐ ClipX
☑ ClipY
☑ ClipZ
☐ flipX
☑ flipY
☐ flipZ
Opacity
Quality
☐ Enable Interpolation
☐ Enable 2D Mode

Figure 5.2: Overview of the multivariate volume visualization interface: (A) dynamic projection panel; (B) volume visualization panel; (C) subspace navigation panel; (D) data operation panel. (A-1) displays a chosen subspace view of the parameter space. (A-2) and (A-3) display the source view and target view during animated transitions between them.

initial volumetric dataset. These approximated cluster centers serve as the representatives of the entire volume and are explored and manipulated during the TF design process. The sampling operation is optional as it ensures reasonable efficiency of the clustering algorithm, in the case of a large volumetric data. The clustering is essential in bridging the gap between processing large number of points in the volume and maintaining interactivity with a small number of points using dynamic projections.

Given a reduced dataset, subspace clustering [29] is applied to represent the high-dimensional parameter space as a collection of low-dimensional linear subspaces. The dimension and basis of each subspace are estimated. I then use these bases to define different viewpoints, that is, I create different projections onto pairs of basis vectors and generate a set of 2D views from each subspace. I further explore these subspace views through dynamic projections in the next stage of the pipeline. During subspace clustering, subspace labels are assigned to the representatives (i.e., approximated cluster centers), and associate points in the entire volume with their corresponding (approximated) cluster centers. Color assignments to the cluster centers then translate directly to color assignments to all points in the volume during the TF design process.

To demonstrate the robustness of the data reduction techniques, I compare the PCA projections of the representative points for different sampling rates and clustering configurations. As illustrated in Figure 5.3, I show the PCA results of a hurricane simulation

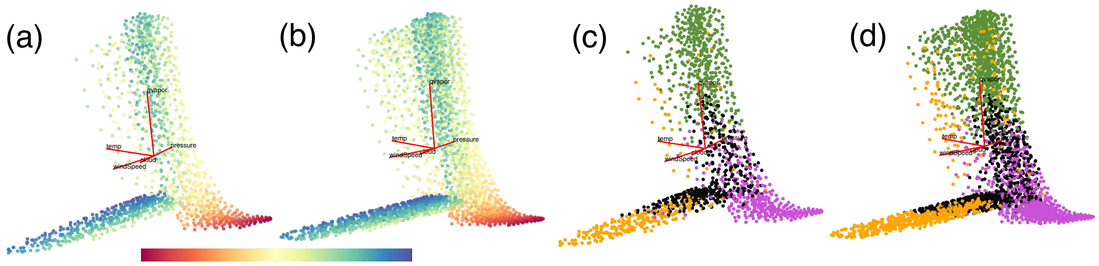


Figure 5.3: Hurricane dataset: comparing PCA results with 1500 ((a) & (c)) and 3000 clusters ((b) & (d)). (a)-(b) PCA projections of the representative points, colored by temperature using the Spectral colormap where red indicates low and blue indicates high values. (c)-(d) PCA projections colored by subspace labels.

dataset (see Section 9.4 for details) using a 10% sampling rate and 1500 clusters (a)-(b) vs. a 40% sampling rate and 3000 clusters (c)-(d). The PCA projections are consistent in terms of point distributions and the subspace clustering results closely resemble one another. This result also shows that 1500 clusters with 10% is sufficient in approximating the structure of the parameter space and its subspaces.

5.3 Transfer Function Design

In this work, a dynamic canvas is utilized as the TF design space. Following data reduction, subspace clustering is applied to identify clusters that shared the same intrinsic low-dimensional subspace in order to capture the structure of the data. In particular, for visualization purposes, each subspace produces several 2D views of the data by creating projections onto pairs of its basic vectors (e.g. a 3D subspace produces three 2D views). These views are then organized in the view navigation panel as illustrated in Figure 5.2(C). Then, dynamic projections are utilized to smoothly transition between different views for parameter space exploration. The animated transitions between these subspace views (and between subspace views and the PCA view) allow the user to gain an intuitive understanding of the structure of the parameter space, for effective TF design.

I provide an illustrative example that contains a few snapshots of such an animated transition in Figure 5.4 from a PCA view to another subspace view, for the hurricane dataset. The points from the blue subspace gradually separate from the rest of the points in the other highlighted subspaces during the animated transitions, which reveals the insight that seemingly connected points in the PCA view may in fact form separate structures.

Based on the information learned from exploring the parameter space, the user can design the colormap by interacting with the points in the dynamic projection panel in multiple subspace views to shape the final TF. The major advantage of designing in multiple

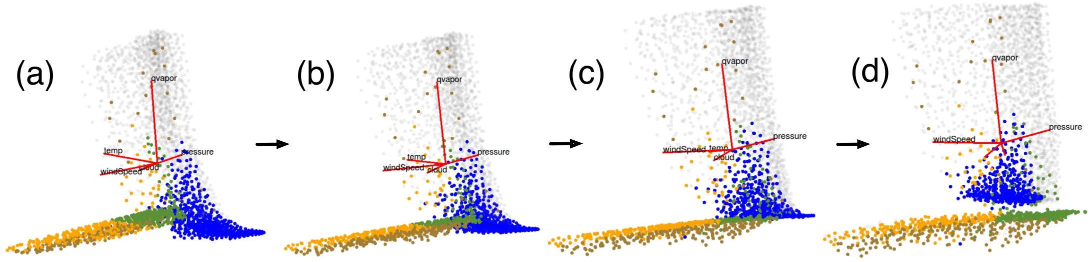


Figure 5.4: Hurricane dataset: animated transitions between its PCA view (a) and a subspace view (d) reveal unseen structures of the parameter space compared to a single static PCA view.

dynamic views over a single static view is the ability to reduce errors caused by structural illusions in a single projection. Due to the high dimensionality and complex nature of the parameter space, typically there is no single projection that faithfully represents the structural relations among its points. With multiple subspace views and the animated transitions among them, the user could start to understand how points from different parts of the data are structurally related to one another.

I provide two approaches for selecting points in the dynamic projection canvas. Lasso selection is used for painting points and sculpting the desired TF regions in multiple views. Neighborhood selection paints a point together with its neighbors within a chosen radius in the high-dimensional parameter space. As illustrated in Figure 5.10, for the hurricane dataset, the user can paint the points via neighborhood selection by increasing the radius on-the-fly, and interactively visualize the corresponding volumes.

Depending on how much the user utilizes the subspace labels associated with the representatives in the dynamic projection panel, three approaches are provided for TF design. The *automatic TF design* assigns colors based on subspace labels and gives a coarse visualization of the volumetric data. This TF assignment is based on the assumption that subspace clustering captures more structural information compared to Euclidean distance based clustering. Subspace highlighting is enabled when dynamically transitioning from a source view to a target view. The points in the subspace that the target view belongs to increase their opacities, whereas all other subspaces increase their transparency. Such a design yields a smooth color transition within the volume visualization during dynamic projections.

The *semi-automatic TF design* is guided by subspace labels and allows more refined TF design. The user can import some or all of the subspace labels as an initial design and then exploit relations among different parts of the data via animated transitions among multiple views for TF modification and refinement.

Finally, during the *manual TF design*, the user utilizes what he or she learned about the parameter space through dynamic projections to manually sculpt the colormap across multiple views. Manual TF design treats the subspace labels as latent information, therefore, allow the user exercises more freedom and flexibility in the design process.

5.4 Implementation and Scalability

Both sampling and k-means++ clustering are implemented in C++. For the subspace clustering and basis estimation, Python is used for faster prototyping and reliable matrix operations. The volume visualization is built on top of an existing high-dimensional data exploration infrastructure (C++ and Qt) where additional modules are introduced to handle multivariate TF design and volume rendering. Several processes are closely related to the issue of system scalability. k-means++ (for subsampling) has a complexity of $O(nk^2)$ times the number of iterations, where n is the number of data points. Subspace clustering incurs $\mathrm{O}(m^2d)$ time and $\mathrm{O}(m^2)$ memory to construct the affinity matrix where m and d denote the number of samples and the data dimension respectively. In addition, hardware capacities limit the size of the raw volumetric data to be rendered, as well as the number of points that could be used to guarantee smooth animated transitions during dynamic projections. I primarily rely on data reduction via sampling and clustering to process large datasets. The sampling rate is maximized as long as the clustering algorithm terminates in reasonable time.

5.5 Application Examples

5.5.1 Hyperspectral Image Dataset

As a proof-of-concept example, the first dataset comes from earth remote sensing using a hyperspectral imaging system. Such a system gathers and processes information collected on an image plane from across the electromagnetic spectrum. It divides the spectrum into a large number of wavelength ranges that go beyond what is visible to the human eye. The dataset has a resolution of 1924 × 753 with a total size of 1.2GB. It is derived from the Moffett Field dataset, part of the AVIRIS Standard Data. A total of 206 wavelengths within the spectrum are selected. Therefore each location in the image corresponds to a point in the 206D space. For data reduction, a 50% sampling rate and 3000 clusters are used. Although this dataset is not volumetric, it can still be used to demonstrate the usage and versatility of the system.

During subspace clustering, a configuration with eight 2D subspaces is arrived. Using automatic TF design (based on subspace labels), four of these subspaces correspond to distinct regions in the image with meaningful interpretations. As illustrated in Figure 5.5, the blue subspace (c) corresponds to the body of water, the yellow subspace (d) contains the urban environment with man-made infrastructures such as roads and bridges, the black subspace (e) represents certain types of buildings (e.g., airports) and the green subspace (f) is likely the vegetation. Such a visualization demonstrates in principle that subspace clustering gives a crude visualization that agrees with intuition or the prior knowledge of the data.

With semi-automatic TF design, a more refined visualization is obtained, as illustrated in Figure 5.6. The user starts by using the black subspace labels (a) to guide the selection of the seed point for neighborhood selection that leads to the magenta area (b) that corresponds to certain types of buildings (notice in particular the striped pattern near a large magenta area, i.e., the airports). Then the user imports the blue subspace labels (c) as they correspond to the body of water almost perfectly. Subsequently, the user chooses a point near the blue area and perform neighborhood selection to arrive at the green region (d) that encloses mostly vegetation. Notice the white grid-like pattern which corresponds to unexplored region with man-made infrastructures. Finally, the user performs neighborhood selection in the projection view seeded from an unlabeled point and arrive at the final yellow region (e) that highlights other types of buildings in the image.

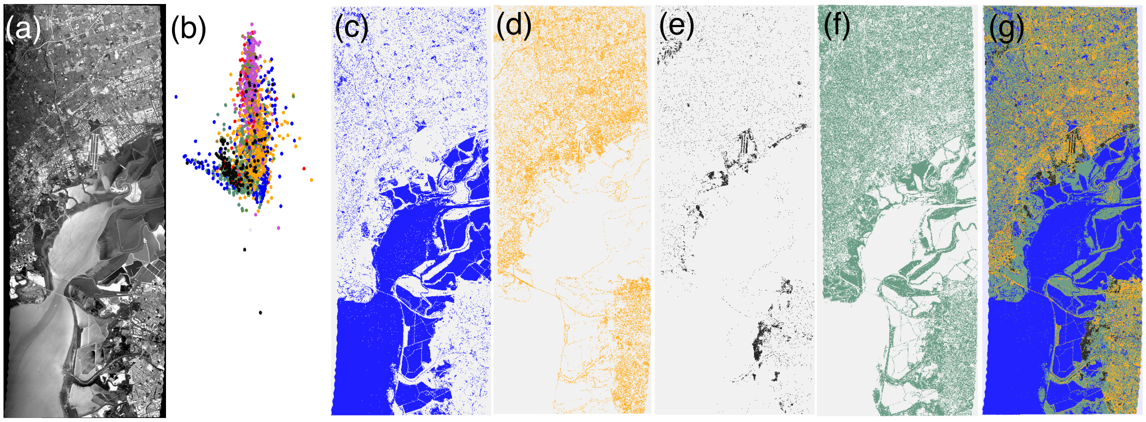


Figure 5.5: Hyperspectral image dataset: automatic TF design. (a) Geographic image of the Moffett Field as a reference point. (b) A subspace view with points colored by subspace labels. (c)-(f) Volume visualizations that correspond to blue, yellow, black and green subspaces, respectively. (g) The combined visualization based on these four subspace labels.

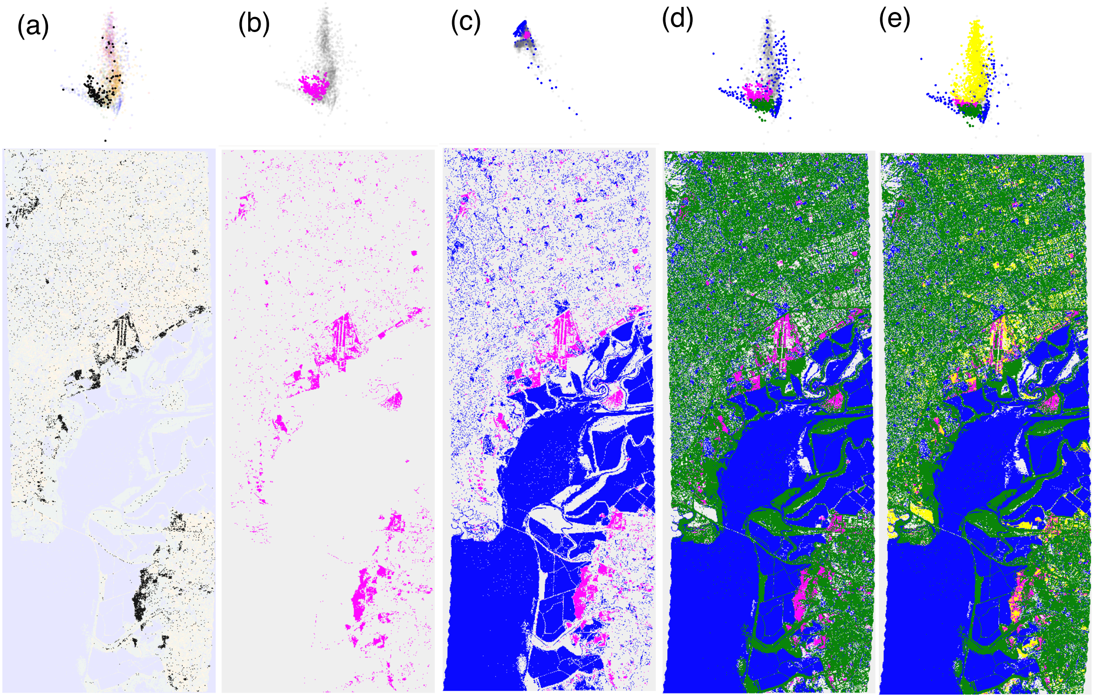


Figure 5.6: Hyperspectral image dataset: semi-automatic TF design. (a) use the black subspace labels to guide the TF design for the magenta area in (b) via neighborhood selection. (c) import the blue subspace labels without modifications. (d)-(e) refine the TF further via neighborhood selections.

5.5.2 Hurricane Isabel Dataset

This dataset originates from a simulation of a hurricane (in particular, Hurricane Isabel from September 2003) from the National Center for Atmospheric Research in the United States. It has a resolution of 500 × 500 × 100 (600 MB), which corresponds to a physical scale of 2139km (east-west) × 2004km (north-south) × 19.8km (vertical). To form the multivariate testing data, each location in the data is mapped to a 6D space (similar dimensions have been used in [122]) by choosing six scalar variables in the simulation including: cloud (cloud moisture mixing ratio), precipitation (total precipitation mixing ratio), vapor (water vapor mixing ratio), temperature, pressure and wind speed. For data reduction, a 30% sampling rate and 3000 clusters are used.

Subspace clustering gives an initial configuration with eight subspaces, which include four 2D subspaces (blue, dark green, black and purple), three 3D subspaces (orange, brown and red) and one 4D subspace (grass green). Such a configuration corresponds directly to an automatic TF design, as illustrated in Figure 5.7, where the PCA views of the parameter space colored by both subspace labels as well as temperature are also included. The blue subspace is shown to contain points with minimum temperature, and the separation between

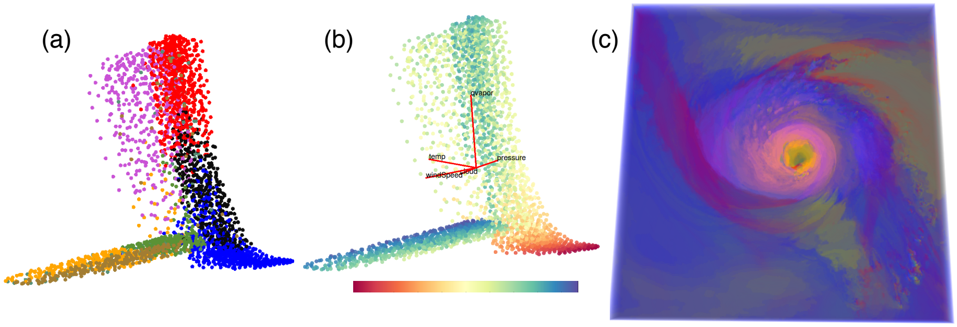


Figure 5.7: Hurricane dataset: automatic TF design based on all subspace labels. The PCA views of the parameter space are colored by (a) subspace labels and (b) temperature. Here the "spectral" colormap is used where red means low and blue means high temperature. The corresponding volume visualization is shown in (c).

the red and the purple subspaces seems to be aligned with the difference between their temperature profiles.

Via dynamic projections, each subspace is highlighted when it transitions into its corresponding views and correspondingly, the user can observe a dynamically varying TF in the volume visualization, highlighting different features captured by each subspace. In Figure 5.8, the user starts from the blue subspace and then dynamically transition to the dark green, brown, red, black, purple, orange and finally grass green subspaces. Based on such subspace exploration, the user sees that sporadic spiral-like features are visible in the brown subspace, which corresponds to a low vapor region, indicated by the direction of the qvapor variable axis in the biplot. On the other hand, the red subspace has relatively high vapor based on the biplot. In addition, the red, orange and green subspaces all cover some parts of the hurricane eye.

Now the user proceeds with semi-automatic TF design, where the existing subspace classifications and the dynamic transitions between different views is utilized to better understand the high-dimensional parameter space. As illustrated in Figure 5.9, starting from the PCA view in (a), the user notices that the orange, green, brown and blue subspaces are intermingled with one another. The user now imports these four subspaces to get an initial TF shown via the volume visualization panel in (b). During dynamic projection, when transitioning from the PCA view to a brown subspace view in (c), the user observes that the blue cluster becomes separated from the rest of the points, whereas the other three clusters remain mixed (the snapshots of such an animation are shown in Figure 5.4). This result demonstrates the effectiveness of the subspace clustering in identifying distinct substructures. The user further explores the relations among these four subspaces via animated transitions in dynamic projections. When transitioning between two orange

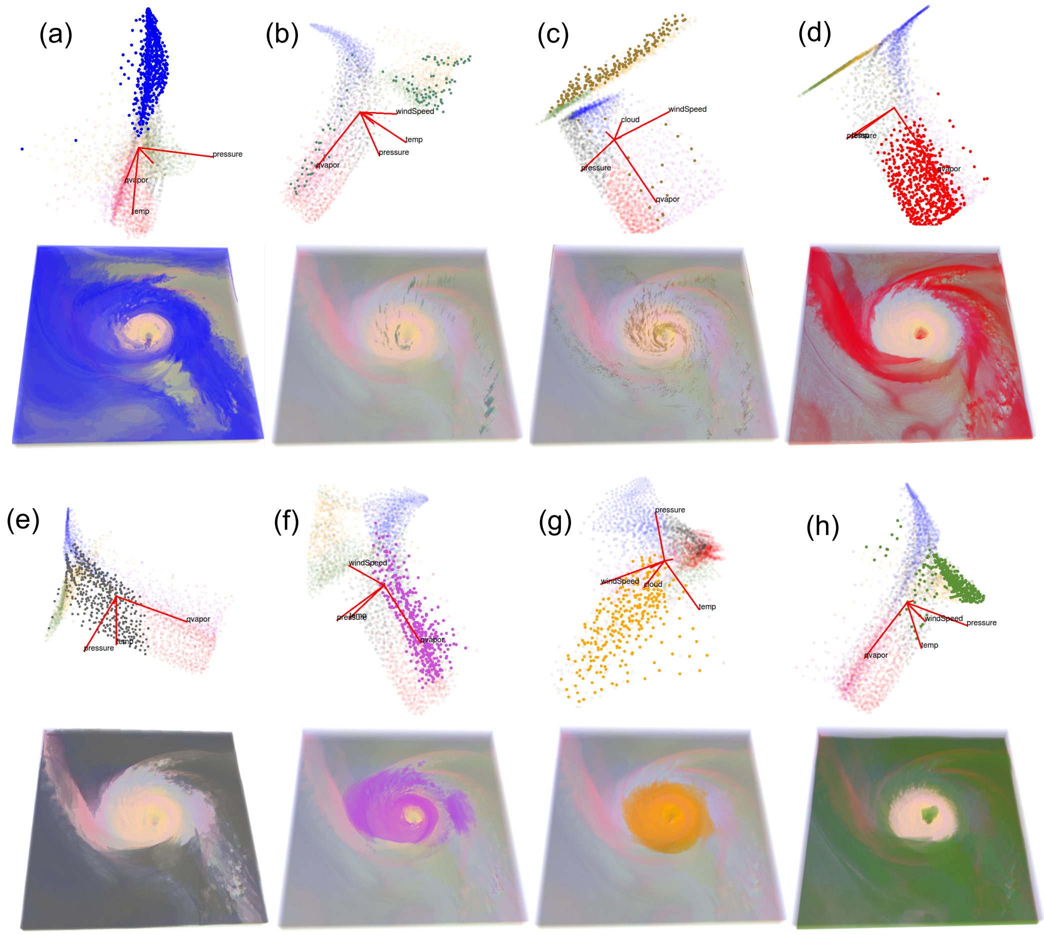


Figure 5.8: Hurricane dataset: automatic TF design. A dynamically varying TF in the volume visualization using subspace highlighting is illustrated. The subspace view navigation panel is showed on the left. Arrows connecting the nodes indicate the current exploration path, which transitions from selected subspace views (a) to (h).

subspace views (d) and (e), the user notices a distinctive spike-like structure pointing towards the opposite direction of the pressure axis in the biplot in (g). By painting such a protruding triangular area with red in (f), a region in (g) that contains the hurricane eye is located.

The user further examines the biplot axes in (f) and notice that the pressure and the wind speed axes point away from the red region, which indicates that the red region has low pressure and low wind speed. Furthermore, the temperature is another dominant axis in this subspace view. To further explore the internal structure of the hurricane eye, the user divides the area into four parts colored by red, yellow, cyan and magenta in (h). Based on the cutaway view in (j), the user sees that the cyan area corresponds to the center of the hurricane eye and the magenta, yellow and red areas form its outer layers. Based on the

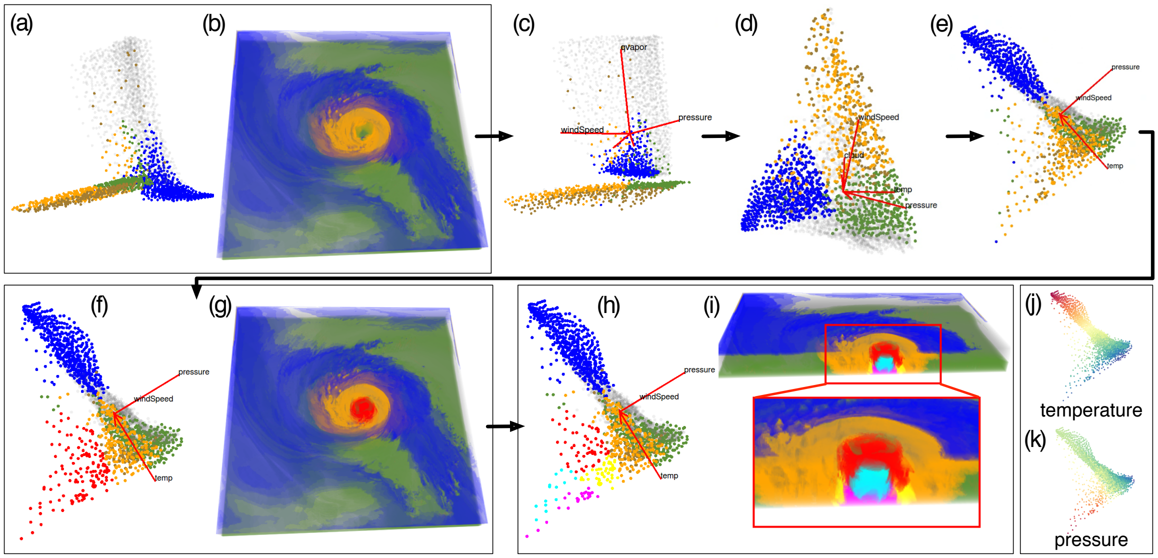


Figure 5.9: Hurricane dataset. Semi-automatic TF design.

relative positions of these four areas and the biplot in (h), together with the temperature (j) and pressure (k) profiles, the user can conclude that the red area and yellow areas have higher pressure whereas the magenta and yellow areas have higher temperature, when compared with the cyan area.

Besides TF design using Lasso selection, the effect of high-dimensional neighborhood selection is demonstrated. As illustrated in Figure 5.10, starting from an initial seed point, the user gradually increases the size of its neighborhood in the high-dimensional

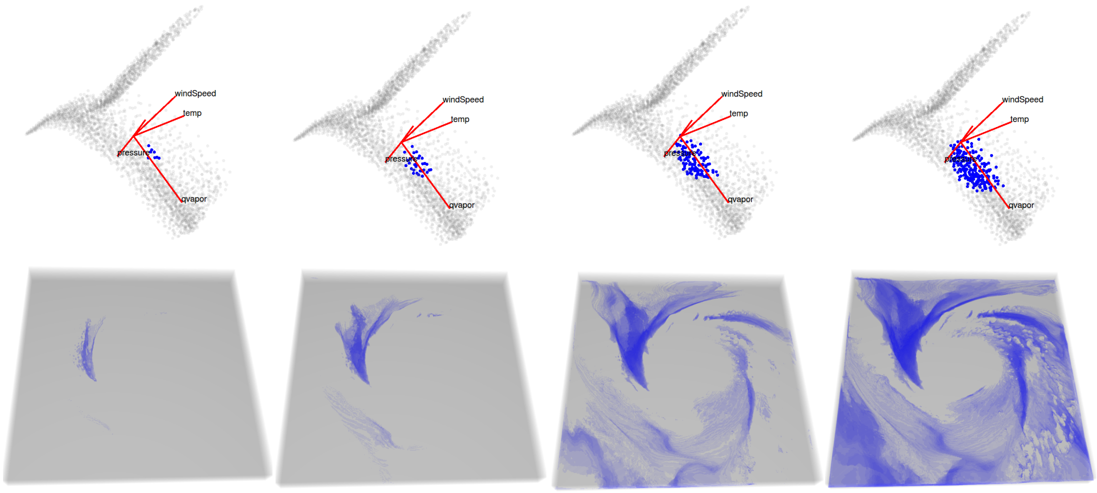


Figure 5.10: Hurricane dataset: showcase neighborhood selection during the TF design as the neighborhood radius increases from left to right.

parameter space and generate a dynamic TF interactively, revealing interesting structures in the volume visualization.

Finally, an example of manual TF design with only Lasso selections is given, as illustrated in Figure 5.11. Via dynamic projections, Lasso selections allow “sculpting” in multiple views of the data during the design process, which touches regions in the high-dimensional space not necessarily reachable by a single static 2D view. The user starts with the PCA view of the data within the dynamic projection panel, by painting with green along a stratified set in (a), which does not correspond to any identifiable structure in the volume visualization. By transitioning between multiple views across different subspaces, the user identifies that the rightmost green area in (b) (roughly enclosed by the black circle) appears to reside in a very different subspace than the rest of the green points. Removing these points (c) results in a void surrounding the hurricane eye (not shown in the figure). The remaining green

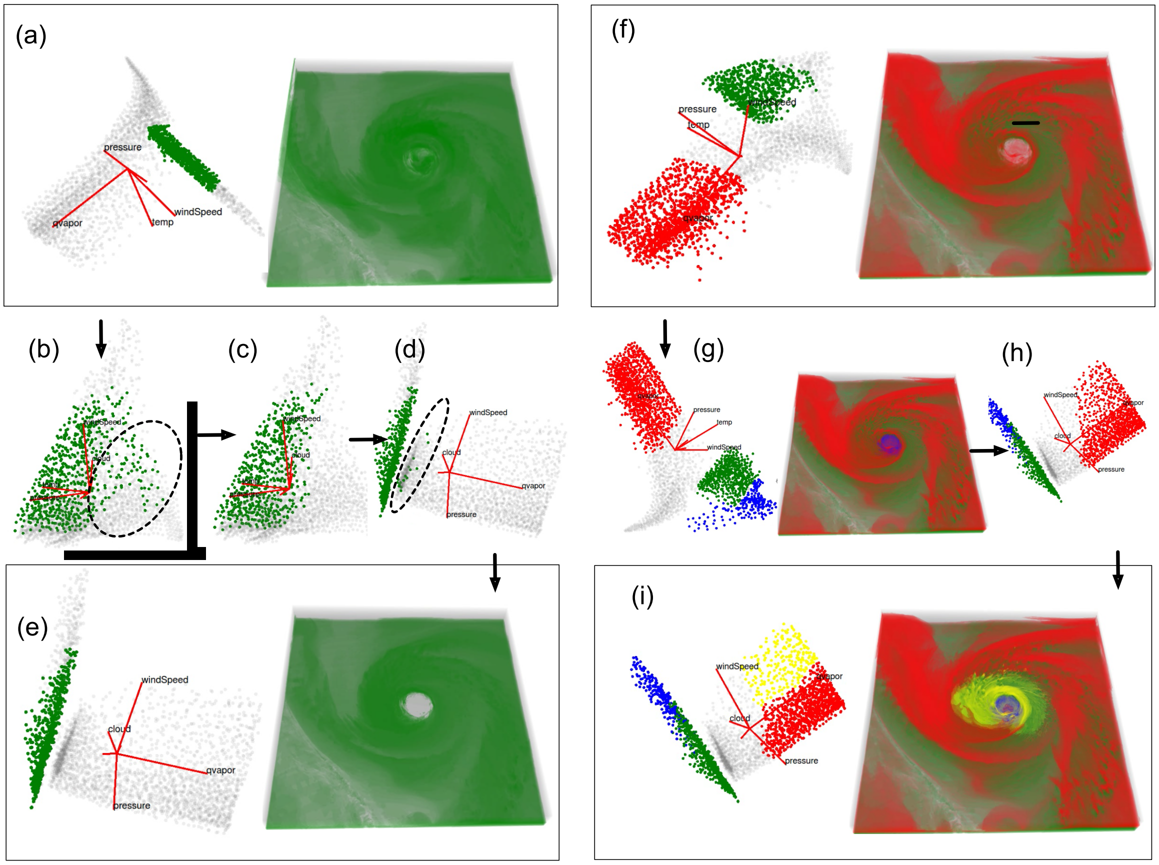


Figure 5.11: Hurricane dataset: manual TF design where the TF is created in multiple views to fully exploit the advantage of a dynamic canvas. (a) Initial design with Lasso selection in the PCA view. (b)-(e) Further sculpting of the TF by removing subsets of green points based on information obtained through dynamic projections. (f)-(i) Performing Lasso selections in different regions of the parameter space for further TF design and refinement.

points split into two clusters when transitioning to a third subspace view (d), and further "sculpting" by removing the small green cluster (enclosed by the black circle in (d)) leads to the visualization in (e) where the emptiness surrounding the hurricane eye becomes readily visible. Subsequently, the user performs Lasso selection in different regions of the parameter space and arrive at a visualization that highlights the hurricane eye across multiple layers along the polar axis.

5.5.3 Ionization Front Instability Simulation Dataset

This data is from an ionization front instability simulation [124, 125]. Scientists are interested in understanding the formation of galaxies, in particular, the effect of "shadow instabilitie", where radiation ionization fronts scatter around the primordial gas. The dataset contains eight chemical species including H_2^+, H_2, H^-, He^{2+}, He^+, He, H^+ and H, as well as a few attributes that measure physical properties, including particle density, temperature and the curl calculated from the simulated velocity field. It has a resolution of $600 \times 248 \times 248$ (1.6GB), and each location is mapped to a 11D parameter space. For data reduction, 25% sampling rate and 3000 clusters are used.

Such a dataset can be described with seven 2D and one 3D subspaces, via subspace clustering. The user starts the visual exploration with the automatic TF design, as illustrated in Figure 5.12. During dynamic projections, the user notices the transformation angles between pairs of subspaces are generally small and no drastic deformation of point cloud exists within most transitions. These transitions indicates a high-level of similarities among the basis vectors describing each subspace. The point cloud in most views forms eclipsed moon like structures and spreads along certain dominant directions. Such a directional pattern is most visible for the red subspace in (a) where the point cloud is elongated into a stick-like structure.

To further explore such a directional pattern, the user proceeds with a manual TF design by assigning different colors parallel to the dominating direction, starting from the subspace view of Figure 5.13(a). The user then transitions to a different subspace view in (b)-(c), proceed further with the TF design and showcase the zoomed-in volume visualization. When transitioning among different subspace views in (e)-(f), the relative positions among groups of points with the same color remain consistent most of the time. Such an observation further validates the simplistic nature of the parameter space for the dataset. In addition to mostly linear transitions among 2D subspace views (in (e)) during the dynamic projections, certain twists and turns are observed among animated transitions between views in the 3D subspace in (f), which provide additional structural understanding of the parameter space.

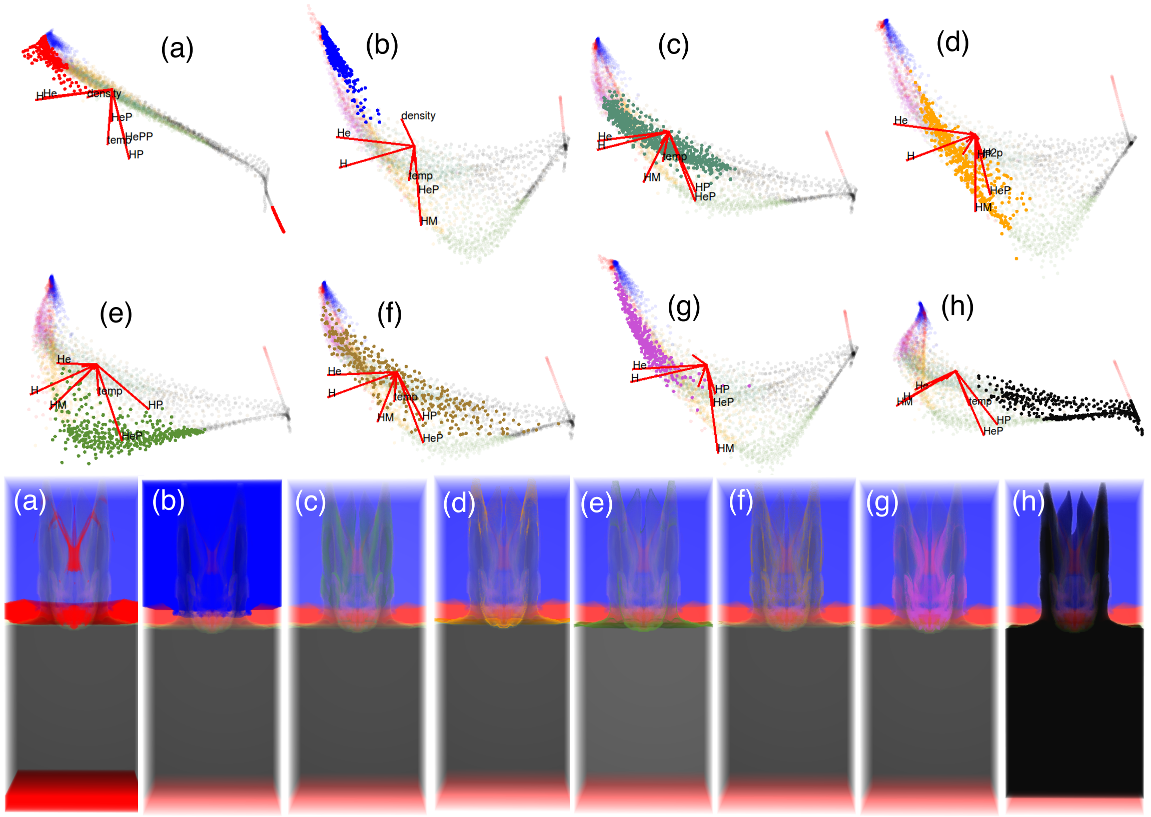
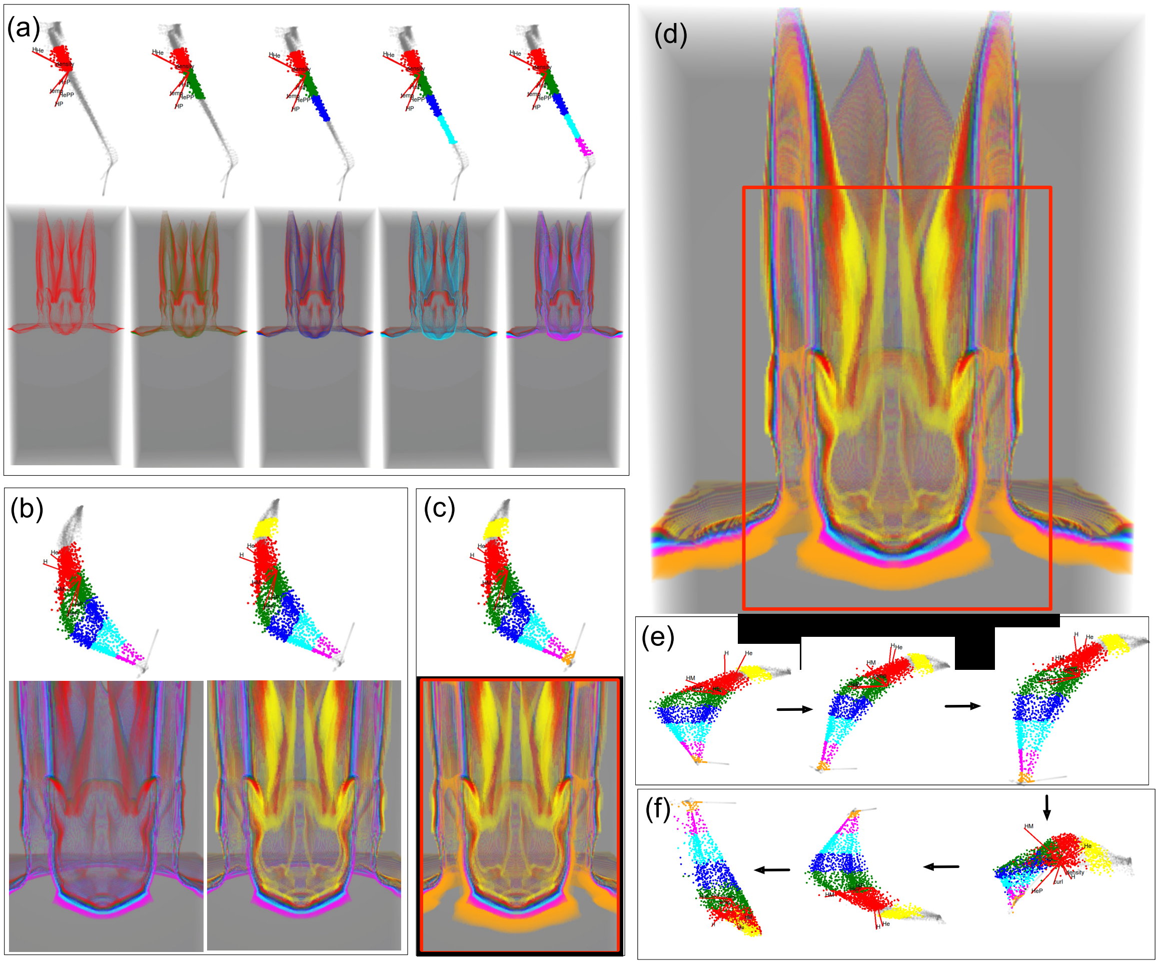


Figure 5.12: Ionization dataset: automatic TF design. (a)-(h) When transitioning between views from different subspaces, a dynamically varying TF in the volume visualization is shown.

The final volume visualization result in (d) displays the layered structure of the ionization front, where the orange outer layer is followed by inner layers in purple, cyan, blue, green, red and yellow, respectively. This pattern is consistent with the chosen colormap along the dominant direction.

As a final note, the TF design for all datasets is carried out on a desktop machine equipped with an Intel Core i5 2.6GHz CPU, a NVIDIA GTX570 GPU and 8GB of memory. For the preprocessing, the running time for sampling from the volume is negligible and k-means++ takes between 30 minutes to 3 hours depending on the size of the data and the rate of convergence. Subspace clustering and basis estimation running on MATLAB take less than 5 minutes. Finally, interactive volume rendering speed (more than 15 frames per second) can be achieved with an appropriate voxel sampling rate.



[redacted]

Figure 5.13: Ionization dataset: example of a manual TF design. By coloring the parameter space along the dominant direction, the user reveals the layered structure within the ionization front. (a) The user assigns different colors parallel to the dominating direction of the projection. (b-c) The user transitions to a different subspace view that expands the design region with further TF modifications. The zoomed-in volume visualization is shown to highlight their fine details. The final visualization is shown in (d). (e)-(f) The relative positions among groups of points with the same color remain consistent most of the time during animated transitions among multiple views.

CHAPTER 6

STUDYING ANALOGY RELATIONSHIPS IN WORD EMBEDDING SPACE VIA SUBSPACE ANALYSIS

Embedding words in a vector space has been a longstanding practice in the natural language processing (NLP) research community. Algorithms, such as Google word2vec [126] or Glove [127], compute the embedding based on large volumes of training articles and the resulting vector space is assumed to encode their semantic relationships. The most notable examples are analogy pairs, such as (king:queen) and (man:woman), where in the appropriate vector space one finds (king + woman - man) $\approx$ queen [128]. In general, encoding words or even paragraphs into intermediate vector representations provides the foundation for a range of different analysis approaches and has been successfully adopted for various applications in NLP. Despite its central importance to the field and wide-scale adoption as a technique, the word embedding space remains opaque to most NLP researchers. Most often it is used as a black box representation for subsequent tasks without an in-depth or intuitive understanding of its structure.

In order to distinguish all the words that exist in a large corpus of text, the embedded dimension is usually chosen to be relatively high, typically between 50 and 300 dimensions. Understanding such a high-dimensional space is an extremely challenging task. Currently, the most commonly used approach by NLP researchers is generating 2D embeddings using t-SNE [59], which result in a single nonlinear projection of all words. Although t-SNE has the ability to embed a large number of words in 2D, it is typically used only as a rough visual representation of overall embedding quality, or to quickly validate computational results. The problem is that due to the nonlinear nature of the embedding, the interpretation of its 2D projection result is challenging. Furthermore, many interesting properties and relationships, such as analogy pairs, are linear in nature and thus will inevitably be lost in the nonlinear embedding.

According to my collaborators in NLP, measuring the analogy pair relationship is the primary method for evaluating word embeddings. These word embedding methods claim geometric relationships (such as the vector relationship in analogy pairs) encode the semantic information between words. Due to a lack of other meaningful ways to compare different word embeddings, the geometric relationships exhibited in analogy pairs have been used to evaluate these claims [128]. Within each analogy group (e.g., a set of analogies that fit the relationship male:female), the vector relationship is checked for every two analogy pairs. For example, the distance between (*king* - *man* + *woman*) and *queen* is taken into consideration for evaluating the embedding quality. The smaller these distances are, the better quality a given word embedding is. However, many open questions remain regarding the analogy relationships and the word embedding evaluation approach. Are there different trends within the analogy group? Do these trends correspond to more subtle but explicit meanings? More importantly, does evaluating word embedding quality via analogy pairs even make sense?

In this chapter, techniques from high-dimensional data visualization are adopted to address these open questions, which provide new insights for NLP researchers. Many properties of interest to the NLP community, such as word analogies, correspond to linear relationships in the ambient space. To highlight these linear relationships, subspace analysis (introduced in Chapter 4) that identifies informative 2D projection is adopted. A novel projection-finding approach that is tailored for the word analogy relationship is also introduced. Compared to the widely used dimensionality reduction strategies in NLP such as t-SNE, the proposed approach better captures many innate linear relationships (e.g., trends within a given analogy group) that are crucial for understanding the analogy relationships in the word embedding space. In order to build a system that provides new capabilities while being user-friendly for NLP researchers, I have worked closely with domain experts and go through several iterations of the design-implement-feedback loop to identify the challenges as well as the right visualization approaches to address them.

6.1 High-Dimensional Word Embedding Space

In this section, background knowledge that is necessary to understand the application domain is discussed.

6.1.1 Word Embedding Algorithm

Word embedding techniques are used to build an intermediate representation of text for subsequent analysis in natural language processing (NLP). Recent approaches, such as word2Vec [126] and Glove [127], have found widespread adoption. The general idea behind word embeddings can be described as follows (see Figure 6.1): Assume we have a dictionary

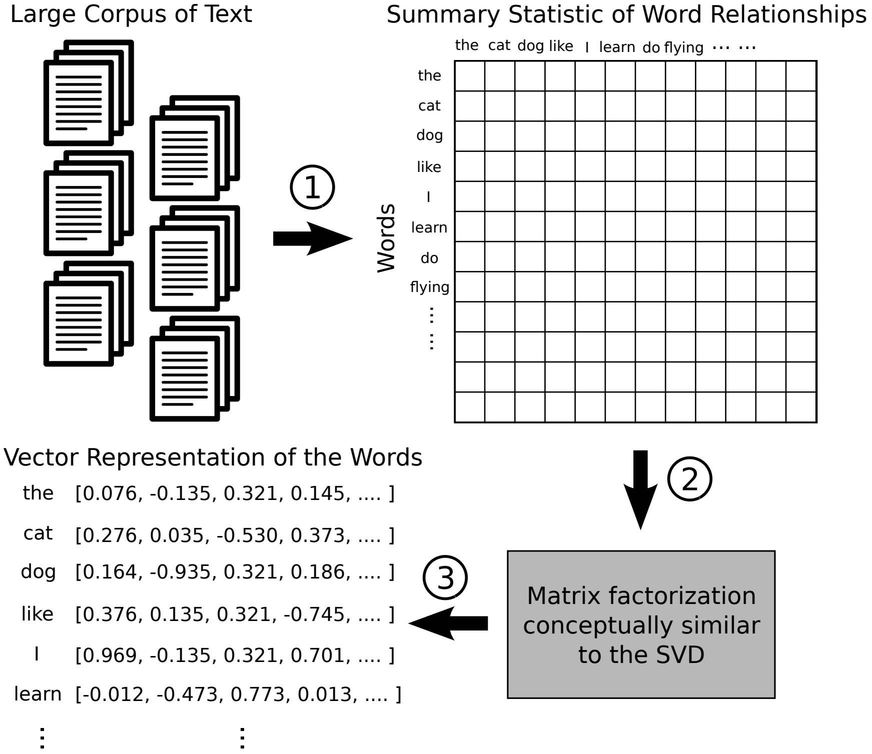

Large Corpus of Text
Summary Statistic of Word Relationships
the cat dog like I learn do flying ⋯ ⋯
the
cat
dog
like
I
learn
do
flying
⋮
Words
1
2
3
Matrix factorization conceptually similar to the SVD
Vector Representation of the Words
the [0.076, -0.135, 0.321, 0.145, ...]
cat [0.276, 0.035, -0.530, 0.373, ...]
dog [0.164, -0.935, 0.321, 0.186, ...]
like [0.376, 0.135, 0.321, -0.745, ...]
I [0.969, -0.135, 0.321, 0.701, ...]
learn [-0.012, -0.473, 0.773, 0.013, ...]
⋮ ⋮

Figure 6.1: An illustration of the word embedding process: The input of the algorithm is a large corpus of text that is summarized in an $n \times n$ matrix M that encodes the relationships between n unique words. Typically, $M(i, j)$ records a statistical relationship, such as the probability of joint occurrence between $word_i$ and $word_j$. Subsequently, M is factorized, and the coordinates in the $d \ll n$ most significant components define the vector representation of words.

of n words $\{w_1, \dots, w_n\}$. One can use the statistical relationships from large text corpora to infer dense vector representations for words, such that words that are semantically connected will reside in proximity to each other. Typically, the resulting vector space has a much lower degree of freedom, i.e., $d \ll n$. Let us denote the vector representations for the words as $w_i \in \mathbb{R}^d$.

More interestingly, with methods such as word2Vec [129] and Glove [127], the embedded words exhibit some surprising algebraic properties, the most notable example being the king:queen, man:woman analogy pairs, where *king* - *man* + *woman* is approximately equal to *queen* (illustrated in Figure 6.2).

Computing word embeddings involves complex statistical and machine learning models beyond the scope of this dissertation. For completeness, I provide a simplified view (based on the word2Vec computation procedure) and refer interested readers to the relevant publications for details [129, 127]. The input to the algorithm is a large corpus of text (e.g., all articles in Wikipedia). The first step is to summarize all pairwise word relationships from

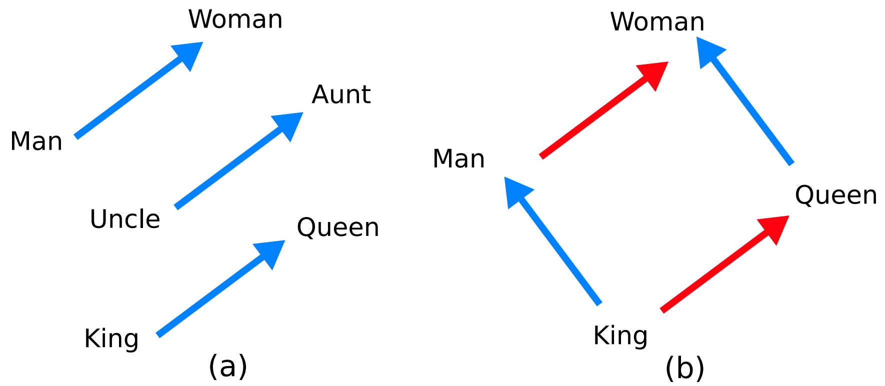


Figure 6.2: In the word embedding space, the analogy pairs exhibit interesting algebraic relationships.

the articles into an $n \times n$ matrix M describing n unique words. An entry $M(i, j)$ records the statistical relationship between $word_i$ and $word_j$. Intuitively, one might consider the frequency of $word_i$ and $word_j$ to appear in the same sentence, although the real model is more complex. More commonly, the pairwise mutual information (PMI) statistic is used. Subsequently, M is factorized in a step conceptually similar to a Singular Value Decomposition (SVD), and the coordinates corresponding to the d largest components are used to represent words.

Understanding high-dimensional spaces has always been a challenging task. In order to distinguish the large amount of vocabulary that exists in the text corpora, words are usually embedded in a rather high dimensional space (50-300), making direct visual exploration very challenging. Furthermore, word embeddings are typically used as an intermediate representation for later analysis. Therefore, users often treat word embeddings as a black box representation, without an in-depth understanding of the relationships among the words in high-dimensional space.

6.1.2 Word Embedding Visualization

Currently, the t-SNE [59], a nonlinear dimension reduction technique, is by far the most common approach for visualizing word embeddings. The t-SNE generates a 2D embedding for a given high-dimensional data. It is ideal for a quick validation of the computation or to obtain a rough estimation of the embedding structure. However, due to the nonlinear nature of the method, many important linear relationships, such as the example illustrated in Figure 6.2, are inevitably lost. From visual analysis aspect, the relationships between analogy pairs relationships can be more effectively visualized using linear projections (e.g., principal component analysis) in comparison to nonlinear approaches such as t-SNE. As

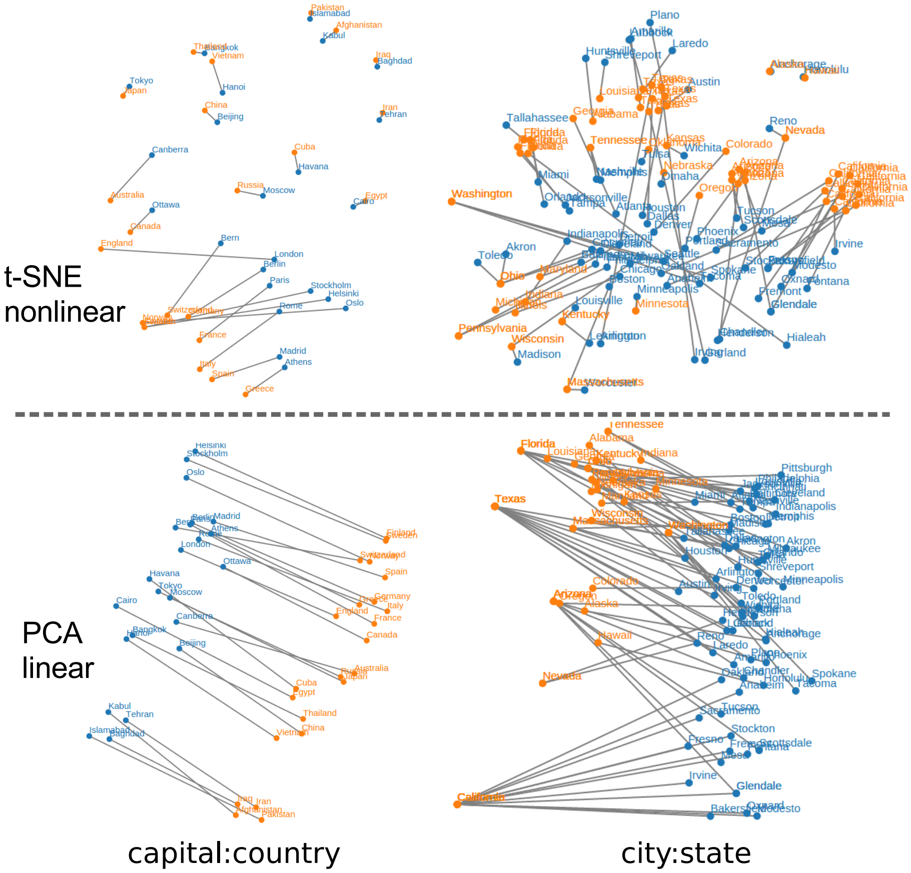


Figure 6.3: Nonlinear embedding methods such as t-SNE are poorly suited for visualizing the relationship of analogy pairs. The orange and blue labels correspond to the two words in a given analogy. The lines indicate the analogy pairs.

illustrated in Figure 6.3, in which the orange and blue colors indicate the two entities of a given analogy and the link connects the words belonging to the same analogy, the trend of analogy pair is totally lost in the t-SNE embedding, while the PCA preserves the apparent relationships. Furthermore, the nonlinear nature makes it impossible to interpret either distances or axis which severely limits the ability to derive insights or meaningful conclusions from the plot. Finally, putting large numbers of points in a 2D visualization can be counter-intuitive. Figure 6.4 illustrates an example visualization with t-SNE, where all the words of interest are shown together in the same visualization. As it can be observed, the individual words are cluttered in the visualization, which provides little information beyond a general sense of colors and neighborhoods.

Despite these limitations, as suggested by the domain scientists, t-SNE is still considered as the de facto standard for visualizing word embeddings (e.g., used for methods comparison in the work [130]). According to the collaborators, no dedicated visualization system exists

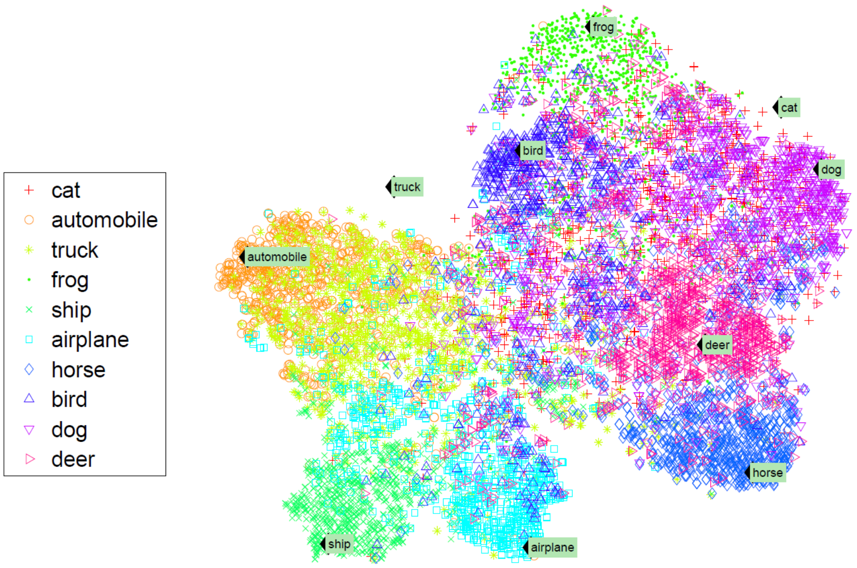


Figure 6.4: An example of t-SNE visualization [2] with different semantic groups indicated by different colored symbols. The t-SNE embedding can provide a rough estimation of the overall distribution of words. However, the visualization is cluttered by individual words, and the words belonging to different semantic groups can be heavily intermingled.

for visualizing word embedding spaces. In this work, I aim to bridge this gap by providing an effective tool specifically designed for visualizing analogy relationships in word embeddings.

6.1.3 Application Goals

The ultimate design goal of the proposed framework is to build a usable and extensible system that can aid the domain scientists in gaining new insights for the word embedding space, and answering specific questions they encounter in their research. The proposed system aims at understanding analogy relationships due to their importance in evaluating different word embedding methods.

In particular, through the interactions with collaborators I have identified three visualization goals revolving around understanding analogy relationships: 1) Identify trends in analogy relationships; 2) Interpret the orientation of the analogy vector direction; 3) Compare the behavior of analogy pairs in different word embedding methods.

These goals aim to answer the following questions. Are there different trends within a given analogy group? Do these trends correspond to more subtle but explicit meanings? Does evaluating word embedding quality via analogy pairs even make sense? Answering these questions are essential for an in-depth understanding of the analogy relationships in high-dimensional space, which, according to my collaborator, is not available before. Finally, visualization techniques can generate visual summaries to provide useful benchmarks for domain scientists interested in a direct comparison of the current plethora of word embedding methods.

6.2 Subspace Analysis for Understanding Analogy Pairs

As discussed previously, obtaining an in-depth understanding of analogy pairs' behaviors is an essential part of studying a word embedding space. In this research, a linear projection based visualization component is included for studying analogy pairs (showed in Figure 6.5(a)). The linear PCA projection helps the user identify the overall trend in each analogy group. However, relationships within each analogy group is not always coherent and separate trends exist within each analogy group (e.g., *noun:plural-noun*). The domain experts are interested in finding these sub-trends (Design Goal 1) that can be emphasized using multiple, complementary linear projections.

In order to address these challenges, the subspace analysis (see details in Section 4), that automatically detect multiple linear subspaces within a given high-dimensional data, is adopted. Subspace clustering methods represent a class of techniques originally developed by the machine learning community. It decomposes the high-dimensional domain into multiple subsets each of which is well contained in a lower dimensional subspace.

For a given analogy group, subspace clustering is applied. And for each of the cluster, an optimal linear projection is generated that best preserves the point relationships within

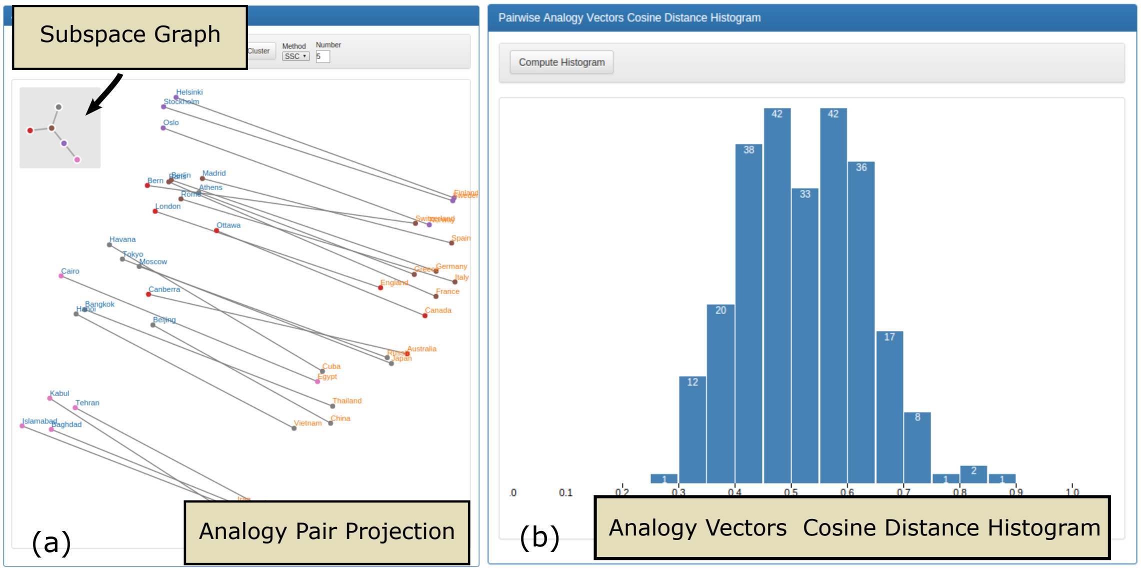


Figure 6.5: Analogy Pairs subsystem subsystem user interface. (a) Analogy Pair Projection produces linear projections of analogy groups. (b) Analogy Vector Cosine Distance Histogram captures the overall coherency of the analogy vector orientations.

the corresponding cluster. To ease the navigation between these selected linear projections, a navigation graph (showed as an inset on the top left of Figure 6.5(a), where each node corresponding to a linear projection) is added to help users navigate among these projections.

6.3 Analogy Orientation Similarity Histogram

Beside identify the trends, domain experts are interested in the orientation of the *analogy vector* (Design Goal 2) constructed as the vector difference of the two words in an analogy (e.g., *man:woman*). In particular, according to the assumption of the word embedding algorithms [128], the analogy vector formed by similar analogy pairs should have similar orientations (i.e., smaller cosine distance). However, to the best of my knowledge, there is no prior work that investigates the distribution of vector orientations in detail, let alone interpret the differences in the orientation of the analogy vectors.

To facilitate such inquiry, histogram of all pairwise cosine distance between analogy vector orientations (see Figure 6.5(b)) is added to the system. This conveys how coherently each analogy group is oriented and whether there may exist additional substructures. In the histogram, the horizontal axis corresponding to the cosine distance, and the larger value is on the right side.

6.4 Projection-Finding Scheme for Analogy Relationship

As demonstrated in the previous section, linear projection approaches, such as PCA and subspace analysis, are very suitable for capturing the linear relationships among the analogy groups. However, a fundamental limitation exists. When generating a projection, all these methods only take the individual words into consideration, without any input or knowledge from analogy pairs relationship. So, strictly speaking, they are not really projecting the analogy pairs.

Therefore, a more desirable method should take the analogy pair information into consideration. In addition, since there is infinite way to project, a more interesting and challenging question is: What is the “best” projection direction for highlighting the analogy relationship?

To address these challenges, in this work, a novel projection method is introduced that tailored specifically for finding the best view to showcase the similarity between analogy pairs. The core idea of this approach is originated from the linear separability of the two concepts in an analogy relationship. As illustrated in Figure 6.6, if a hyperplane is optimized to maximum the separation between the two concepts in an analogy, the normal direction

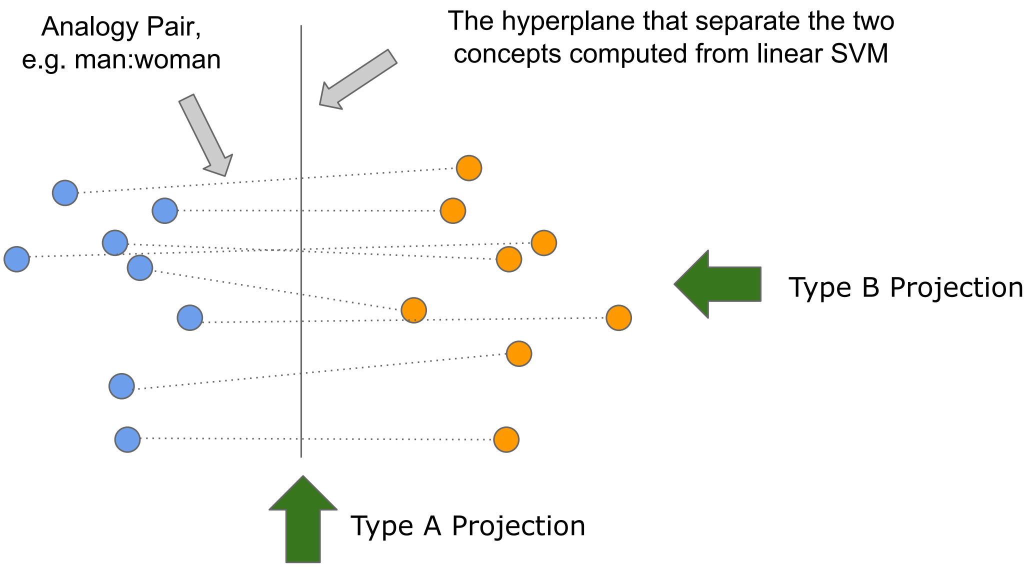

Analogy Pair,
e.g. man:woman
The hyperplane that separate the two
concepts computed from linear SVM
Type B Projection
Type A Projection

Figure 6.6: The domain specific projection finding scheme. The two projection directions complement each other in aid the interpretation of analogy pair coherence.

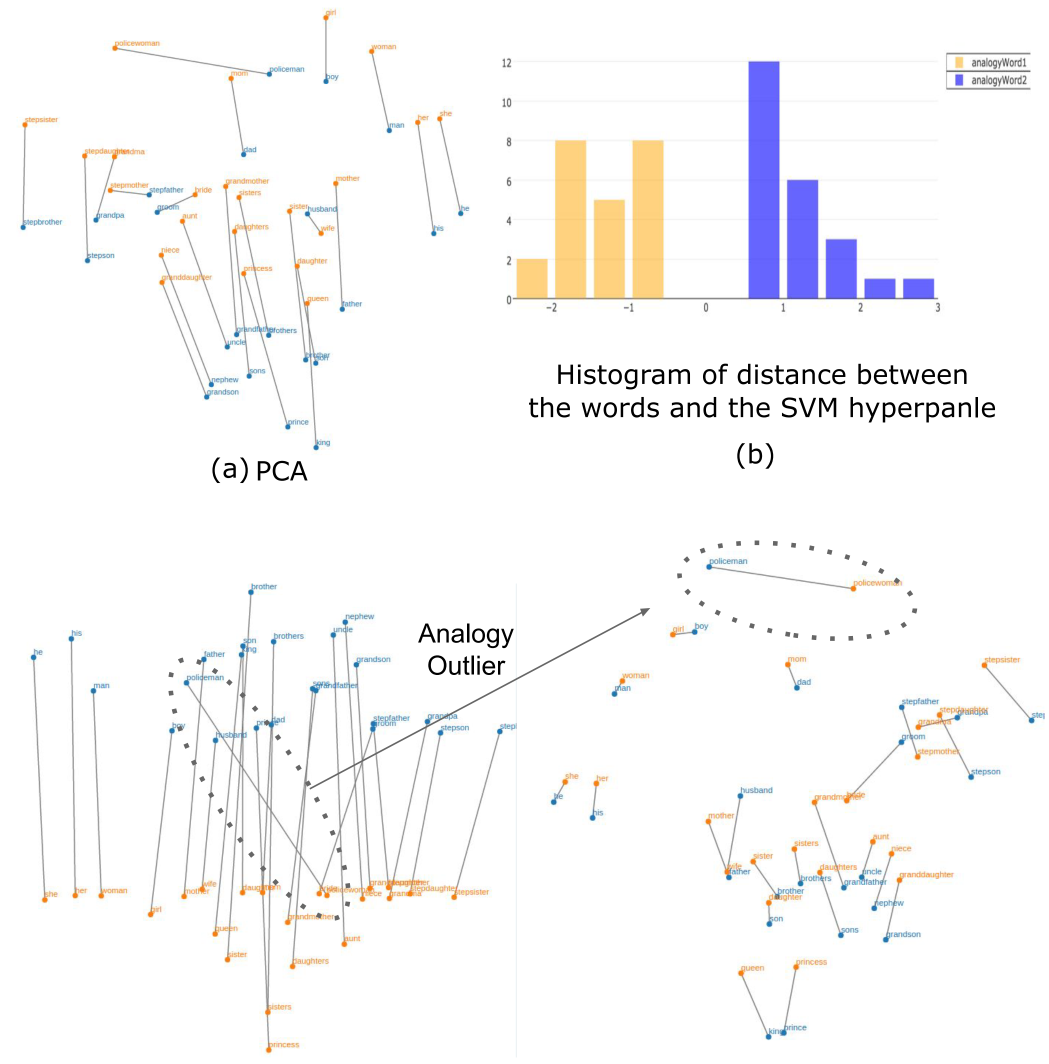
likely will capture the most dominant direction among the analogy vector orientations. Such a hyperplane and it normal can be computed from a linear support vector machine [131]. After finding the normal direction, one projection basis is obtained, and it captures the dominant direction of the analogy. To better showcase the analogy relationship, identify the variation in the analogy group is important. Here, PCA can be used to find the maximum variance in one of the analogy concept to form the second basis. With these two bases, 2D projection along the “side” of analogy pairs can be found (Type A projection in Figure 6.6).

To provide complementary information that highlights the divergence of analogy vector orientations, the pairs can be projected onto the hyperplane (Type B projection in Figure 6.6). The analogy pairs that is much longer than the rest are likely the outliers (very different analogy vector orientation compared to the rest).

As illustrated in Figure 6.7, for the *man:woman* analogy group, PCA (Figure 6.7(a)) can not find the linear projection that highlight the coherency among the analogy pairs. By utilizing the projection finding scheme that tailored for showcase the analogy vector directions, the analogy relationship is easily captured for *man:woman*. The outlier, *policeman:policewoman*, are also easily revealed (Figure 6.7(c) (d)).

6.5 Implementation

As illustrated in Figure 6.8, the system is split into server and client. The server handles complex computation while the client manages the user interface. The web-based client allows me to continuously share the latest developments with my collaborators, which is



(a) PCA

Histogram of distance between the words and the SVM hyperpanle

(b)

(c) Type A Projection

(d) Type B Projection

Figure 6.7: The domain specific projection-finding scheme. The two projections direction complement each other in aid the interpretation of analogy pair coherence.

crucial for a tight design-implement-feedback loop. The communication between the web client and the server is accomplished by a set of RESTful APIs. To achieve a good trade-off between implementation complexity and performance, the server is implemented in *Python*, and the computation methods are handled by efficient libraries (e.g., scikit-learn [132]) or python binding of native C/C++ code. The client graphical interface is implemented in *JavaScript* and *HTML*, *d3.js* [133] is adopted for handling graphical elements. The web-based visualization system is built iteratively following the visualization goals (discussed in Section 6.1.3), and constant feedbacks from the collaborators.

According to the domain scientists, the larger the input text corpus is, the better the quality of the embedding will be. Therefore, all the experiments are carried out with the

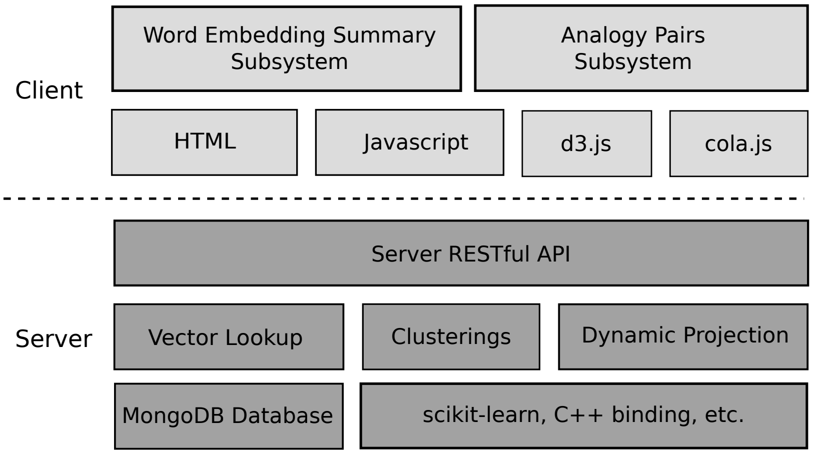


Figure 6.8: The overview of system architecture. The entire system is split into server and client, where the server handles complex computation tasks and the client handles user interaction and display.

established and widely used pre-trained datasets (word2Vec googleNews dataset with 3 million words and phrases, and Glove Common Crawl dataset with 2.2 million vocabulary). These datasets typically contain millions of words, thus I have incorporated the MongoDB database [134] on the server to store and access the pre-trained data. With database in place to fetch the word vectors instantly, combined with efficient implementation of the computation algorithms, all operations can be carried out interactively on the full-scale 300-dimensional pre-trained word embedding spaces.

6.6 Application Results and Evaluation

In this section, the proposed system is tested to address the domain specific questions and providing domain experts with new insights. The dataset used to carry out this study is one of the standard analogy pairs test datasets used in various NLP research. In this dataset, analogies are categorized into multiple analogy groups, each group corresponding to one analogy type, such as, *country:capital*, *verb:past-tense-verb*, *singular:plural*, etc. In all the examples, the 300-dimensional version of pre-trained datasets (Glove or Word2Vec) are used.

6.6.1 Trends in an Analogy Group

The relationships between analogy pairs is essential for studying the properties of the word embedding space. In the following examples, the analogy pairs related questions can be addressed by utilizing different projection-finding methods.

First, the domain scientists are interested in identifying trends within each analogy group (Section 6.1.3). In the proposed system, by utilizing subspace analysis on the analogy group, different trends (each corresponding to a linear subspace), in which similar analogy pairs reside, can be uncovered, Compared to clustering method such as k-Means++ [135], in which pairwise distances are directly used for clustering, subspace clustering identifies groups of point that share similar lower-dimensional subspace independent of the euclidean distance between them.

As illustrated in Figure 6.9(a), k-Means++ clustering will generally group close-by words in term of semantics, but the subspace clustering (Figure 6.9(b)) instead groups related analogy pairs that share similar trends. Instead of relying only on the overall linear projection, multiple linear projections each focusing on a localized trend within the analogy group are presented, allowing the user to view the relationships from multiple perspectives.

As showed in Figure 6.9(c), the opacity of the analogy pairs corresponding how well the distance between the two words in the analogy is preserved. The distance between words in the pink cluster are better preserved by the current projection compared to the rest of the words in this analogy group. In addition, the transitions between linear projections are handled by dynamic projection, where a series of intermediate linear projections are generated to create a smooth and meaningful transition for better tracking of changes.

In Figure 6.10(d)(e)(f), the *adj:comparative* analogy group is showed. Figure 6.10(d) illustrates the overall trend of the analogy group as captured by PCA . With the help of the system, the domain scientists are able to view the analogy relationships from multiple perspectives, where each highlights a localized pattern that otherwise will be suppressed by the dominant overall trend.

6.6.2 Not All Analogies Are Created Equally

Furthermore, that different types of analogy groups can have very different behaviors regarding their coherency. In Figure 6.9 and Figure 6.10, the subspace can highlight trends of analogy pairs. However, exceptions do exist. As illustrated in Figure 6.11(a), (b), (c), the subspace clusters are identified in each of the analogy groups. The dotted circles are added to the figure to highlight cluster patterns. In Figure 6.11(a), *country:nationality*, closely related analogy pairs are grouped into the same cluster (e.g., Scandinavian countries and capitals). In Figure 6.11(b), *currency:country*, the two concepts in the analogy are distinct, so when applying subspace clustering, currencies and countries form their own subspaces. In other words, subspace clustering identifies the stronger linear trends in the currencies and countries, instead of analogy pairs. In Figure 6.11(c), as showed in the overall PCA linear projection, the *singular:plural* analogy group contains words that have very different concepts (*animal:animals* vs. *fruit:fruits*), therefore, when applying subspace clustering,

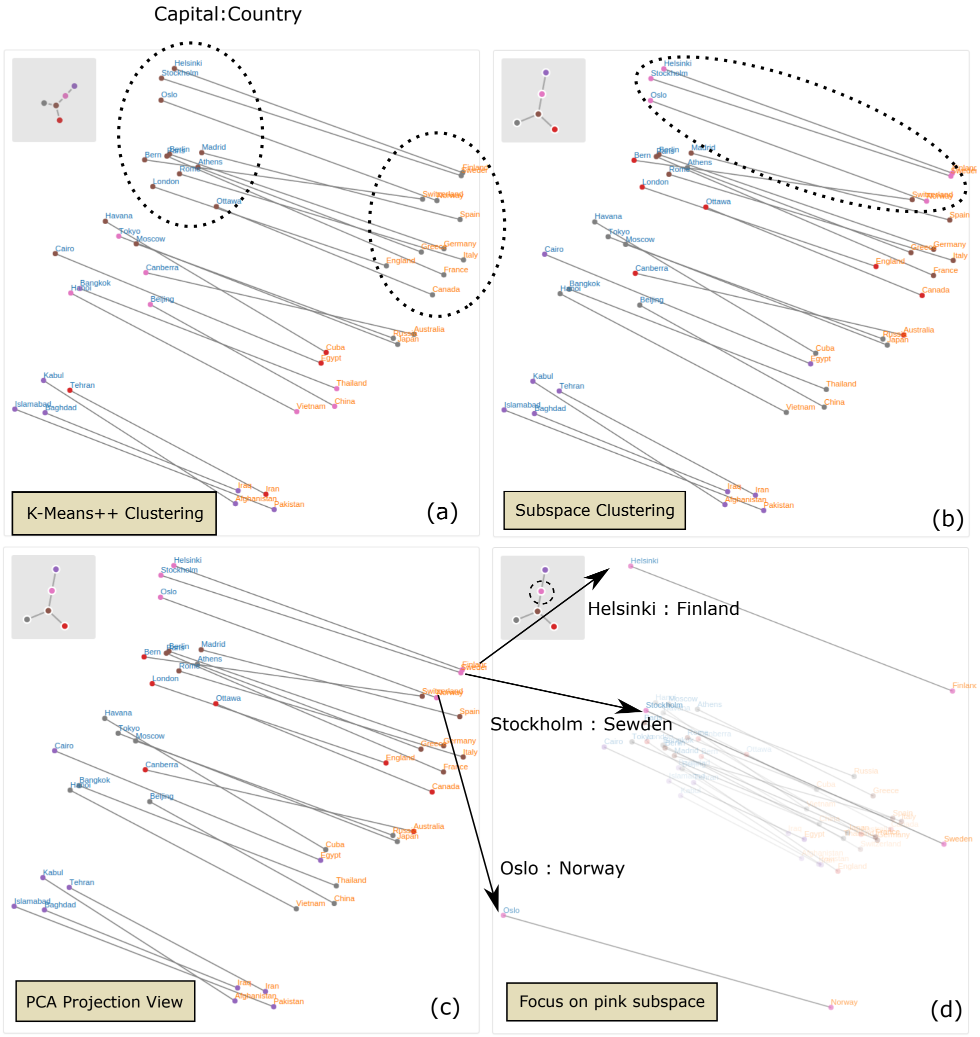


Figure 6.9: Subspace clustering helps identify different trends in an analogy group. (a) k-Means++ clustering result, where close related words are grouped. (b) Subspace clustering partitions analogy pairs based on their analogy relationship's trends. (c) The linear projection that best preserves one of the trends (pink subspace). In (c), the distortion based opacity encoding is enabled, in which the less well-preserved analogy relationship become more transparent.

animal and fruit words are grouped into different subspaces, not necessarily due to the pairs have very similar orientation, but because the differences between concepts have a stronger influence on the subspace distance. By comparing the histogram in Figure 6.11(d), (e), (f), we can see in Figure 6.11 (d), *country:nationality* is the group with the most coherent analogy vector orientations. In Figure 6.11 (f), *singular:plural* corresponds most analogy pairs are quite different orientation. The *currency:country* (see Figure 6.11 (e)), is somewhere in-between.

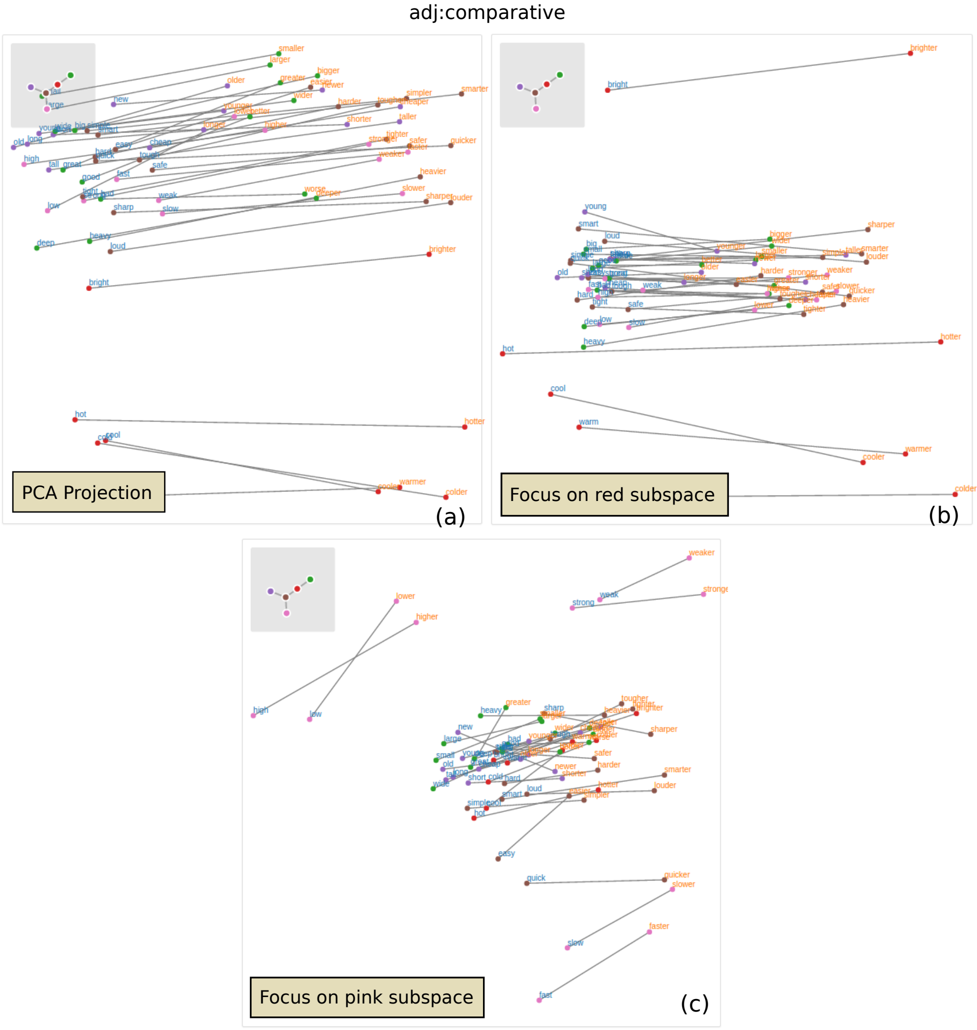


Figure 6.10: In (a)(b)(c), the *adj:comparative* analogy group is illustrated. (a) shows the overall trend captured by PCA. In (b), the projection focuses on the red subspace while (c) explores the pink subspace. With the help of the tool, the domain scientists are able to view the analogy relationships from multiple perspectives that otherwise will be suppressed by the dominant overall trend.

Therefore, if algebraic vector relationships are used in analogy groups as the quality measure (as suggested in [128]), then applying it to different kinds of analogies produces quite different errors estimation. However, the differences are not necessarily due to the embedding quality but due to the innate differences between different analogies, which can be easily observed in the visualization. The collaborators have indicated that the differences between analogy groups have not be discussed in existing NLP literatures, even though such an observation may have a significant impact on how one should approach different analogy groups.

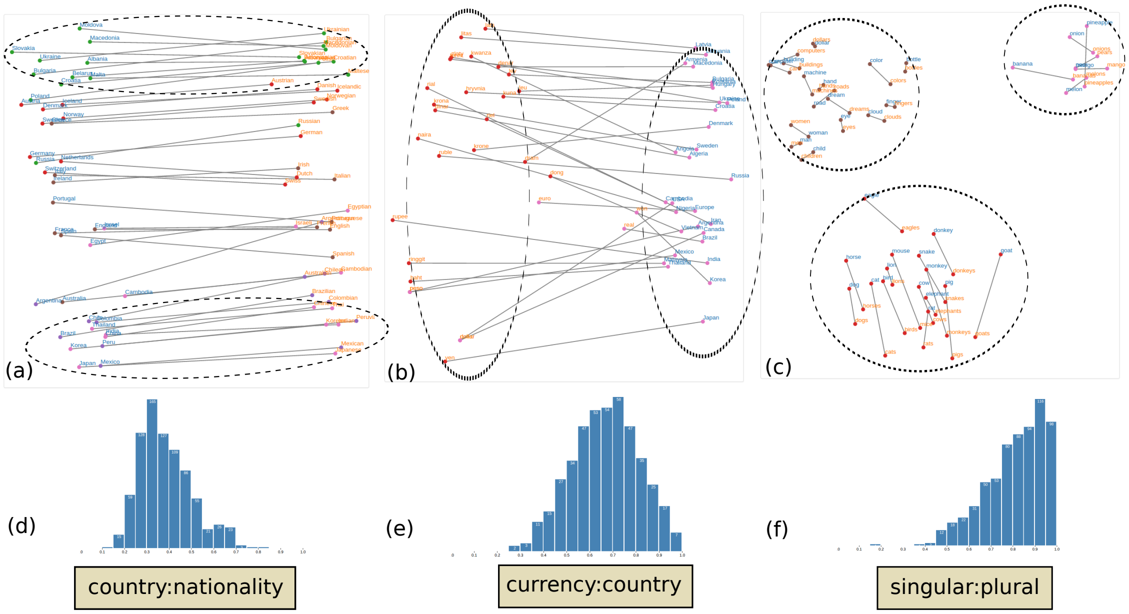


Figure 6.11: Different types of analogies can lead to very different behaviors of analogy pairs. In (a), (b), (c), the subspace clusters are identified in each of the analogy groups. The dotted circles are added to highlight cluster patterns. In (a), closely related analogy pairs are group into the same cluster. In (b), *currency:country*, the two concepts in the analogy are distinct, so the subspace clustering focused on the linear trends within the currencies and countries, instead of among the analogy pairs. In (c), as showed in the overall PCA linear projection, the *singular:plural* analogy group contains words that have very different concepts (animal, fruit, etc.), therefore, analogy pairs are grouped into clusters not necessarily because they share a very similar linear subspace, but due to the larger distances between these different concepts have a stronger influence. The histogram in (d), (e), (f) confirms the observation.

6.6.3 Word Embedding Methods Comparison

The histogram of the pairwise cosine distance provides a summary of the orientation coherency of the analogy vectors. Therefore, as illustrated in Figure 6.12, it can be used to compare the behavior of different word embedding methods. Based on the histogram, we can see the word2Vec perform better for the *currency:country* analogy group, while Glove perform better for the *nationality:country*. But, on average, both methods perform quite similarly.

6.6.4 Expert Evaluation

The system is designed based on the constant feedback from domain scientists. In the beginning of the study, the focus is on designing a better alternative for t-SNE to highlight linear relationships. However, as the collaboration deepened, the importance of the analogy relationships and how they connect to the evaluation of the word embedding quality lead

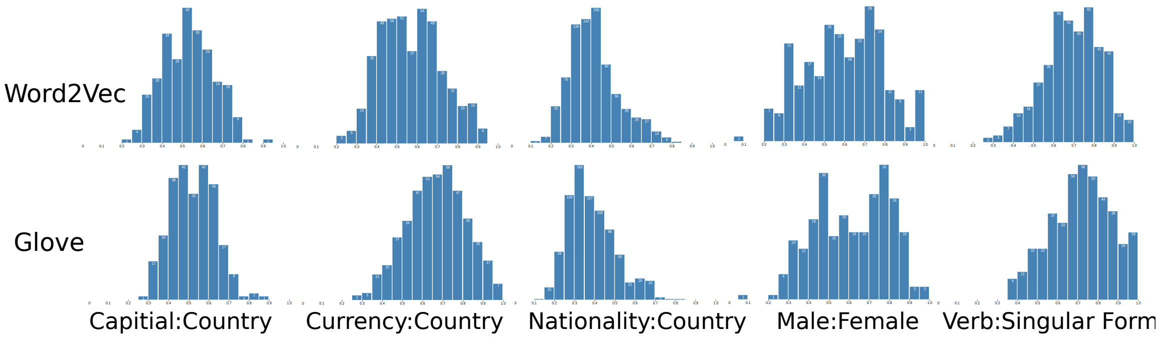


Figure 6.12: Comparison between word2Vec and Glove regarding the pair-wise analogy vector cosine distance distributions.

to the development of a tool that focuses on visualizing analogy relationships. Through multiple designs iterations, the visualization goals are identified and achieved.

First, by utilizing the proposed tool, the domain scientists find similar concepts can form subspaces in the word embedding space (see Figure 6.9). The subspace clustering helps the user obtain multiple projections that highlight different trends within each analogy group. Moreover, the domain scientists have learned from the histogram of pairwise cosine distance that the orientation variations of some analogy groups are unexpected high compared to popular belief, while other analogy groups show relatively coherent orientations. Such an observation leads to a very important finding: the type of analogy will greatly impact the coherency of the analogy relationships, not all analogy relationships are created equally! Existing evaluation approaches for word embedding make the assumption that all types of analogy relationships should be preserved. However, based on extensive exploration via the proposed tool, the domain scientists conclude that for relationships such as noun:plural-noun, where the semantic difference within an analogy (cat vs. cats) is small and the semantic difference among the analogy pairs (cat:cats vs. apple:apples) are big (see Figure 6.11), enforcing the relationship such as “cat - cats + apples = apple” may not make a lot of sense. In other words, the difference in analogy groups should be taken into consideration when using them to measure word embedding quality. Instead of blindly using all types of analogy relationships to evaluate the word embedding, an examination on how to select the “right” analogy relationships is necessary. Finally, the domain scientists indicate that the visualization approach provides a refreshing new perspective for comparing different embedding via visual encoding and summary.

PART III

STRUCTURAL SUMMARY OF THE SPACE OF 2D PROJECTIONS

CHAPTER 7

GRASSMANNIAN ATLAS

The focus thus far in this dissertation has been the identification of a selected set of 2D projections. However, by studying the space of all 2D projections as a whole, new insights can be gained.

Among related works, approaches to find (a selection of) “interesting” projections based on one or multiple metrics have received significant attention. A commonly adopted strategy is described as follows: a large number of different candidate projections can be created by, for example, exploring all possible axis-aligned projections [107, 16] or through random samplings [111, 116]. The candidate projections can then be ranked according to some user-defined quality metrics [136, 43, 137, 1, 34]. In the end, a collection of top ranked projections is presented to the user. This approach has been successful in finding small sets of meaningful projections, but it has some drawbacks. First, except for very restricted cases (i.e., axis-aligned projections of moderate dimension), it is not feasible to explore all possible projections and it is unclear how well a given set of candidates samples such a space. Second, using only a given quality metric, it is difficult to compare different projections and thus determine which are potentially redundant and which represent fundamentally different aspects of the data. Third, the ranked results rely completely on the intrinsic properties of the metric itself, and it is not straightforward to analyze the effect of different metrics. For example, some metrics may naturally emphasize multiple good projections, each of which preserves certain high-dimensional relationships in some subset of the data but distorts the others; whereas some other metrics might be more global and tend to highlight all relationships equally well (or badly). Finally, despite the many metrics that have been introduced, so far little effort has been spent on analyzing the metrics themselves, on comparing their fundamental properties (as opposed to their results), or using such information to guide the selection of projections.

To address these challenges, a general framework [30], referred to as the *Grassmannian Atlas*, is introduced. The *Grassmannian Atlas* captures the global structural variation of a quality metric within the space of all linear projections. The atlas exploits the fact that the set of all linear projections forms a manifold called the *Grassmann manifold* (or *Grassmannian*) [25] with a well-defined geodesic distance metric and of known dimensions (see Section 2.3 for details). The candidate projections are obtained by uniformly sampling on this manifold, expressing a given quality metric as a function defined on this manifold and using tools from scalar field topology to extract its global structure. Furthermore, the concept of topological spines introduced in [26] is adopted to serve as an intuitive interface for users to explore the space of all projections according to a given quality metric. Topological spines use an easily accessible terrain metaphor that naturally groups different projections around local optima of the metric and highlights the relationships among different groups, i.e., how different (in value) and how far apart (on the Grassmannian) their corresponding optima are. Finally, the topological structure provides new insights into the behavior of metrics and an intuitive approach to compare metrics. For example, it is easy to determine which metrics tend to highlight different complementary projections or which require more or less candidate projections to be reliable.

7.1 Overview of the Computation Pipeline

As mentioned above, the Grassmannian Atlas is designed to provide a more intuitive and reliable approach to select a set of 2D linear projections for visualization of a given high-dimensional data. The challenge is that there exist an infinite number of possible projections, and the top ranked ones according to some quality measure may not be the most informative ones. In particular, similar projections are likely to have similar quality measures. Consequently, a cluster of very similar projections will be chosen over a potentially very different and more informative projection with slightly lower ranking.

Instead, this research select a set of locally optimal projections as representatives based on computing the high-dimensional topological structure of the chosen quality measure. Figure 7.1 provides an overview of the approach. First, I randomly sample a (large) set of linear projections represented as linear subspaces. A neighborhood graph can then be computed on these samples to obtain a discrete approximation of the Grassmannian manifold, which defines the space of all linear projections (see Section 2.3). I then evaluate the chosen quality measure on the Grassmannian and compute its topological spine (Section 7.3). The local maxima of the topological spine then indicate locally optimal projections (with respect to the given measure), i.e., those that cannot be improved with incremental changes. Finally, the topological spines also serve as a convenient and intuitive interface to navigate between different projections.

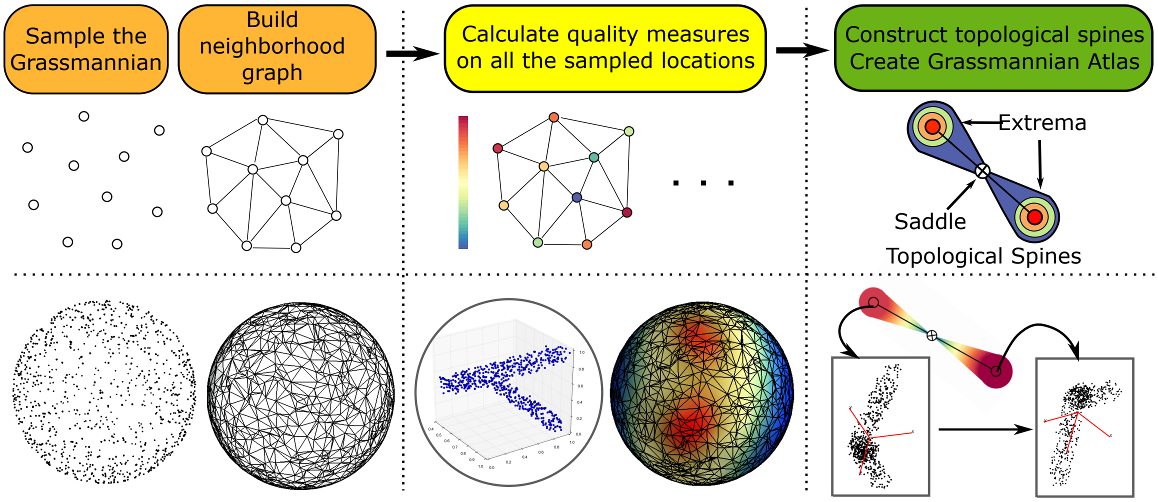

Sample the Grassmannian
Build neighborhood graph
Calculate quality measures on all the sampled locations
Construct topological spines Create Grassmannian Atlas
Extrema
Saddle
Topological Spines

Figure 7.1: First row: the three steps (marked with different colors) for constructing the *Grassmannian Atlas*. Bottom row: examine the space of linear projections involving a 3D example. For illustration purposes, the left panel displays point cloud samples representing projections rather than subspaces as the Gassmannian has no intuitive embedding.

7.2 Sampling the Grassmannian

I model the space of all linear projections based on a Grassmannian that parameterizes all 2D linear subspaces of a high-dimensional dataset, and provide a sampling strategy to approximate the Grassmannian in any dimension.

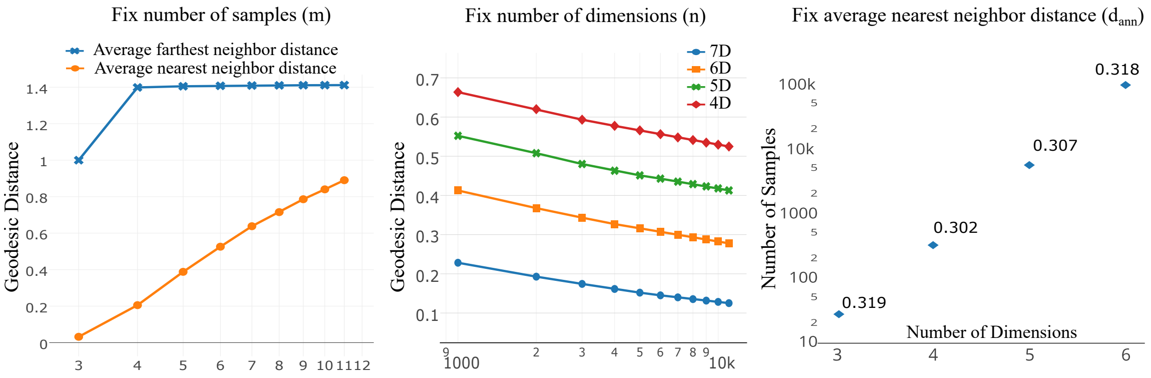
7.2.1 Uniform Sampling

To obtain an approximation of the Grassmannian $Gr(2, n)$, I generate a discrete point sample of the manifold and construct a neighborhood graph based upon the geodesic distances on the manifold. Ideally, the sample should be uniformly random and dense to adequately capture the structure of the manifold, as well as the structure of a reasonable function defined on the manifold. First, how to construct an approximately uniform sampling of a given size is discussed. Later, experiments for understanding the relationships among input data dimension, sample size, and sample density, are provided. The sampling quality is evaluated in Section 7.6.

A random sample on the Grassmannian $Gr(2, n)$ can be generated by constructing uniformly distributed random rotation matrices [138]. More specifically, the QR decomposition [139] of a Gaussian random matrix S (i.e., a matrix that contains random numbers with a Gaussian distribution) is used to compute a random rotation matrix T, that is, $T = Q \cdot \text{diag}(\text{sign}(\text{diag}(R)))$ where $S = QR$. A random sample on the Grassmannian therefore corresponds to a 2D subspace generated by applying a random rotation matrix to a pair of standard basis in $\mathbb{R}^n$. To ensure the set of rotation matrices is approximately uniformly distributed, resample can be applied to the initial points using the k-means++ seed point initialization algorithm [135], which maximizes the spread of points by selecting points away from already selected samples. Finally, a neighborhood graph is constructed connecting the sampled points using geodesics. Since the sample is approximately uniform, a k-nearest neighbor graph (kNN) is sufficient (with an appropriately chosen k). Such a graph is a discrete approximation of $Gr(2, n)$ that supports the subsequent topological analysis.

7.2.2 Sampling Experiments

In practice, for a given data dimension n, the choice of the number of samples is crucial for reliable analysis of the data. To this end, I study the relationships among the number of samples (m), the data dimension (n), and the sampling density defined by the average nearest neighbor distance (d_{ann}). In Figure 7.2(a), for a fixed $m = 1500$, I vary the data dimension n where $3 \leq n \leq 10$, and compute d_{ann}. d_{ann} increases as n grows exponentially (notice that x-axis is log-scale), indicating increasing sparsity in higher dimensions. In Figure 7.2(b), for a fixed n ($4 \leq n \leq 7$), d_{ann} decreases with the exponential increase of m (notice that the x-axis is log-scale). Finally in Figure 7.2(c), I illustrate that for an approximately fixed $d_{ann} \approx 0.3$, the required number of samples m increases exponentially with the number of dimensions n (notice that the y-axis is log-scale).



(a)

(b)

(c)

Figure 7.2: Sampling experiments. Let m be the sample size, d_{ann} be the average nearest neighbor distance, and n be the data dimension. (a) For a fixed $m = 1500$, d_{ann} increases with an exponential increase of n (x-axis, log-scale). (b) For a fixed n ($4 \leq n \leq 7$), d_{ann} (y-axis) decreases with an exponential increase of m (x-axis, log-scale). (c) To maintain a fixed density $d_{ann} \approx 0.3$, m (y-axis, log-scale) scales exponentially with n (x-axis).

7.3 Quality Measures

The proposed framework applies to any quality measure; in this work, I focus on three categories: *scagnostics* [1, 18], *projection pursuit indices* [140, 141], and the measures derived from objective functions of dimensionality reduction methods [31].

The graph-theoretic scagnostics comprises a set of nine measures describing the shape, trend, and density of points from linear projections: *outlying*, *skewed*, *sparse*, *clumpy*, *striated*, *convex*, *skinny*, *stringy*, and *monotonic*. These measures help to automatically highlight interesting or unusual scatterplots from a scatterplot matrix. Scagnostics computation relies on graph-theoretic measures such as the convex hull, alpha hull, and minimal spanning tree of the points. Take the *skinny* measure for example, $c_{skinny} = 1 - \sqrt{4\pi area(A)}/perimeter(A)$, where A indicates an alpha hull of the points in the projection.

Projection pursuit indices are quality measures developed on the basis of the original projection pursuit approach [9] to capture various features in a projection. In particular, I include *gini*, *entropy* [141] (highlighting class separation), *central mass*, and *hole* [140] measures in this study. Finally, the objective functions of dimensionality reduction methods are also used for identifying interesting projections. Linear Discriminant Analysis (LDA) can be adopted to measure the amount of class separation. *Stress*, which is the objective function in the distance scaling version of Multidimensional Scaling (MDS), measures the quality of distance preservation. Let d_{ij} be the distance between a pair of points i, j in $\mathbb{R}^n$ and $\hat{d}_{ij}$ be the corresponding distance in $\mathbb{R}^k$, where $k < n$. Stress is defined as $\sum_{i,j}(d_{ij} - \hat{d}_{ij})^2 / \sum_{i,j} d_{ij}^2$ [142].

Given an approximation of the Grassmannian, I consider various quality measures of interest as scalar functions on the Grassmannian, and calculate their values on all the sampled locations.

7.4 Extract Topological Structures

Given the list of subspace (samples) with the corresponding quality values, the tradition approach simply selects the highest ranking projections and presents them to the user. However, as discussed above, some of these projections may be similar and thus redundant. Consider the 1D example of Figure 7.3(a). The two highest ranking samples are close together, i.e., represent a very similar projection, but the second peak is ignored, even though in practice it may provide a very different projection and thus likely more information. Treating the samples as individual points, these relationships between subspaces are difficult to consider. Exploiting the underlying manifold structure, however, leads to an intuitive definition of locally optimal subspace. Given both the samples and their neighborhood

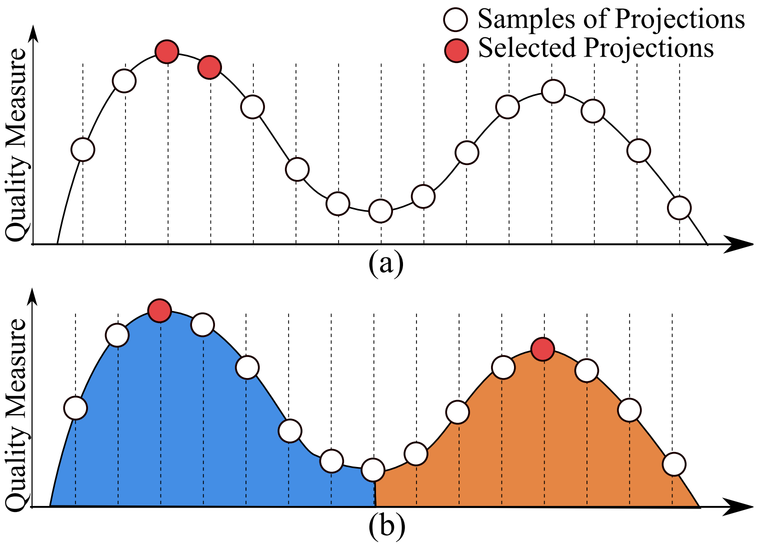


Figure 7.3: Selecting projections based purely on the ranking of a quality measure, (a) fails to identify structurally distinct projections as those obtained via topological analysis (b).

relations, it is natural to consider only those subspaces that have no neighbor with a higher metric value. Intuitively, we prefer projections where no small adjustment could lead to a higher quality value. Such a tendency naturally leads to the concepts of topology and in particular the Morse complex.

7.4.1 Morse Complex and Persistence

I use the topological notions of *Morse complex* to identify local maxima of a function and *persistence* to quantify their robustness.

Given an Morse function defined on a smooth manifold, $f : \mathbb{M} \to \mathbb{R}$, An *integral line* of f is a path in $\mathbb{M}$ whose tangent vector agrees with the gradient of f at each point along the path. An integral line starts at a local minimum and ends at a local maximum of f. *Descending manifolds* (surrounding local maxima) are constructed as clusters of integral lines that have common destinations. The descending manifolds form a cell complex that partitions $\mathbb{M}$, referred to as the *Morse complex*.

In the context of this research, $\mathbb{M}$ is the Grassmannian, a smooth manifold without a boundary, and f is a quality measure of interest. We identify local maxima of f based on the Morse complex, and they correspond to structurally distinct regions within the landscape of f. To further quantify the robustness of a local maximum, the notion of topological persistence is used. The *persistence* of a local maximum is defined to be the minimum amount of perturbation to the function that removes it. In Figure 7.3, for example, the right peak is less persistent than the left peak, since it can be removed with a nearby critical point (e.g., a local minimum) with a smaller amount of perturbation. I use the discrete algorithm of [143] to approximate the Morse complex of a measure, given a sampling and neighborhood graph as discussed in Section 2.3.

7.4.2 Topological Spines

The Morse complex provides a structural summary of the topology of a function, and is well defined in any dimension, but it is not easy to visualize. Instead, the concept of topological spines [26] is used to visualize both the space of projections and an intuitive interface for users to select and explore various projections (see Figure 7.4)

The topological spine adapts a terrain metaphor, as shown in Figure 7.4, that connects local maxima whose corresponding descending manifolds have shared boundaries. Intuitively, these connections can be interpreted as the ridge-lines between neighboring peaks of a terrain. The topological spine uses two parameters to simplify its structure. First, *persistence* is used to remove noise and artifacts to construct a simplified dual complex. Second, a *variation* threshold is provided to determine which of the remaining connections should be considered "ridge-like", and only those above the threshold are visualized. Furthermore, the size of each cell in the Morse complex, i.e., the number of samples it contains, is encoded by the width of the topological spine. The persistence plot (see Figure 7.4) is essential for understanding the distribution of robust features in the function: a long flat plateau indicates the existence of multiple robust peaks that are good candidates for selection, whereas a descending slope suggests excessive noise and the lack of robust structures.

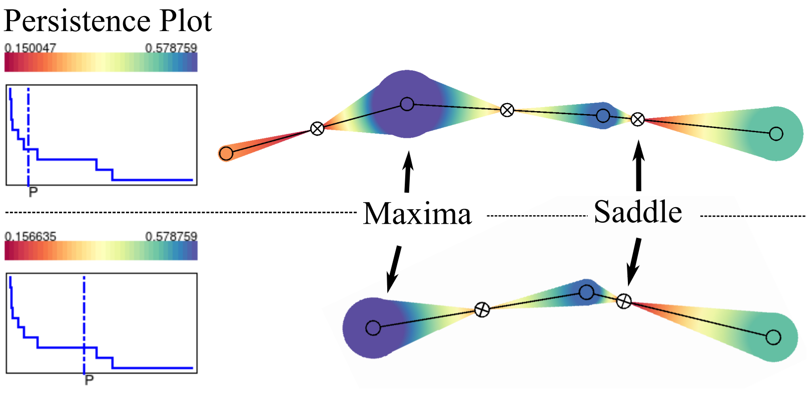


Figure 7.4: Multiscale topological spine representations. The persistence plots are shown on the left: the x-axis corresponds to the persistence threshold, and the y-axis is the number of current cells in the simplification. The long plateau in the persistence plot (bottom) corresponds to a stable topological structure.

Apart from the automatic selection of locally optimal projections, the proposed system also allows users to interactively explore the different local maxima using the topological spine as a selection interface. In particular, the system allows selection of the simplification (persistence) levels that automatically updates the spine and provide dynamic transitions between maxima/projections. The interface consists of two linked views, the topological spine panel and the dynamic projection panel. The former displays the topological spine of the chosen quality measure at the selected persistence set directly via the embedded persistence plot (see Figure 7.4). The projection panel displays the dataset using the currently selected linear projection (local maxima). To better understand the relationships between projections I use the dynamic projection approach [116] to create animated transitions between projections by displaying a set of intermediate linear projections.

7.4.3 Computation Complexity

Since the sampling of Grassmannian $Gr(2, n)$ and the construction of neighborhood graphs are independent from the actual dataset as well as the quality measures, the sampling process need to be computed for each dimension n only once. Let m be the number of data points, n the number of data dimensions, and k the number of samples on the Grassmannian. Evaluating the quality measures for each linear projection takes between $O(mn^2)$ (Scagnostics with binning optimization) and $O(m^2n)$ (Stress). The algorithm used to construct the topological spine from the samples of a given quality measure has a complexity of $O(k \log k)$. Therefore, the overall computation complexity for a given data with a selected quality measure is $O(m^2nk + k \log k)$. The theoretical relationship between the number of samples k and data dimension n is examined in Section 7.6. Quality measures and their corresponding topological spines are pre-computed to support interactive exploration. For the examples discussed in this dissertation, the computation time varies between 2 to 30 minutes, depending on the data dimension, sample size, and the number of quality measures. The test setup consists of a machine with Intel Core i5 2.8GHz processor running Linux. The software framework is written in C++/Qt and compiled with GCC 4.8.

7.5 Validation with Synthetic Data

In this section, the robustness and correctness of the computation pipeline is validated through synthetic data example. I first evaluate the sampling procedure by showing that the proposed approach samples the Grassmannian evenly and completely. Subsequently, I show that the topological structure is stable for different sampling sizes and neighborhood graphs.

7.5.1 Sampling Density and Sampling Size Parameter Validation

To reliably represent functions defined on the Grassmannian, a uniformly distributed sample that covers the entire manifold is required. For moderate input dimensions, the Grassmannian has comparatively low dimensions, and creating sufficient samples, especially during offline pre-processing, is straightforward. If the data dimension becomes too large for the available resources, the Grassmannian has been shown to be amenable to dimension reduction, i.e., a PCA [49].

To validate the results, Figure 7.5 shows the histogram of nearest neighbor distances and farthest neighbor distances for $10k$ samples from $Gr(2,5)$. As expected, the nearest neighbor distances are tightly clustered, indicating a nearly uniform distribution. Similarly, the farthest neighbor distances indicate that the entire manifold has a "diameter" of 1.4. As the sample is random and/or re-sampled, the uniform farthest neighbor distance makes it unlikely (though not impossible) that the manifold is not completely covered. However, a high-quality sample of the Grassmannian does not necessarily guarantee that a given metric defined on the Grassmannian is well sampled.

Figure 7.6 shows the persistence plots and topological spines for the two-planes dataset (discussed in details in the next section) for different numbers of samples and different neighborhood sizes for graph construction. All results are stable, indicating that at least for this dataset the Grassmannian is sufficiently sampled and the proposed approach is

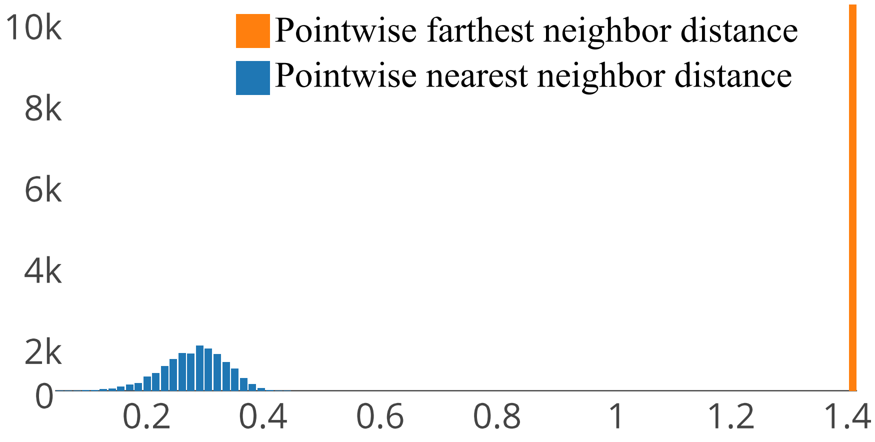


Figure 7.5: A histogram showing the distribution of pointwise nearest (blue) and farthest (orange) neighbor distances for $Gr(2,5)$ with $10K$ samples.

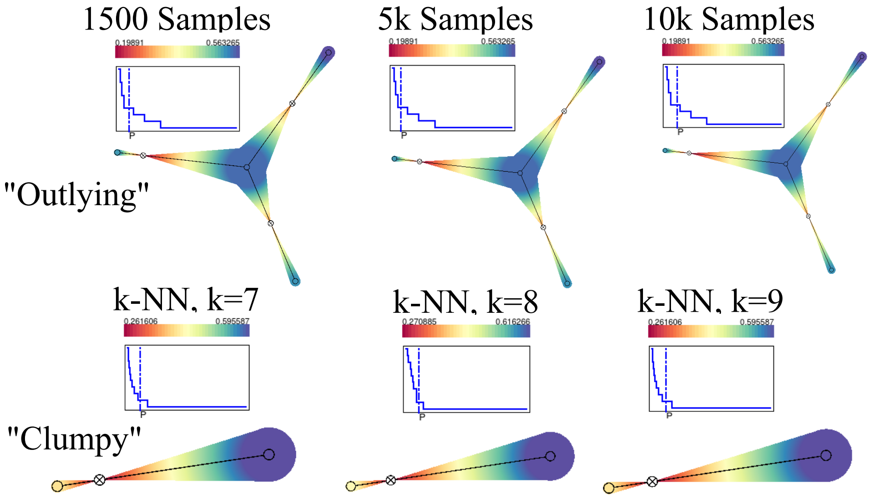


Figure 7.6: Validating the stability of topological spines by varying the number of samples and the number of neighbors for the k-NN graph.

numerically stable. Similar parameter studies are performed for all experiments to ensure the correctness of the results.

7.5.2 Validation with Synthetic Two-Plane Dataset

To evaluate the effectiveness of the proposed approach I analyze a synthetic dataset containing samples from two 2D planes embedded in $\mathbb{R}^3$ that intersect with a 75-degree angle (see Figure 7.7(a)). The *scagnostics skinny* measure (Figure 7.7(b)) identifies the head-on projection in which both planes are skinny as the main mode and various other projections where only a single plane is "skinny" as alternatives. The *Stress* measure (Figure 7.7(c)) finds only a single, stable maximum, which identifies an average projection in which both planes are equally distorted. The projection pursuit index *central mass* (Figure 7.7(d)), on the other hand identifies good projections for both planes as local maxima. These experiments demonstrate that the Grassmannian Atlas not only is able to identify good projections but also provides insights into the measure itself. A measure with only a single stable maximum likely produces some globally average projection whereas multiple maxima indicate several complementary views emphasizing different, local aspects of the data.

7.6 Application Examples

In this section, several real word data are used to demonstrate the effectiveness of the proposed method.

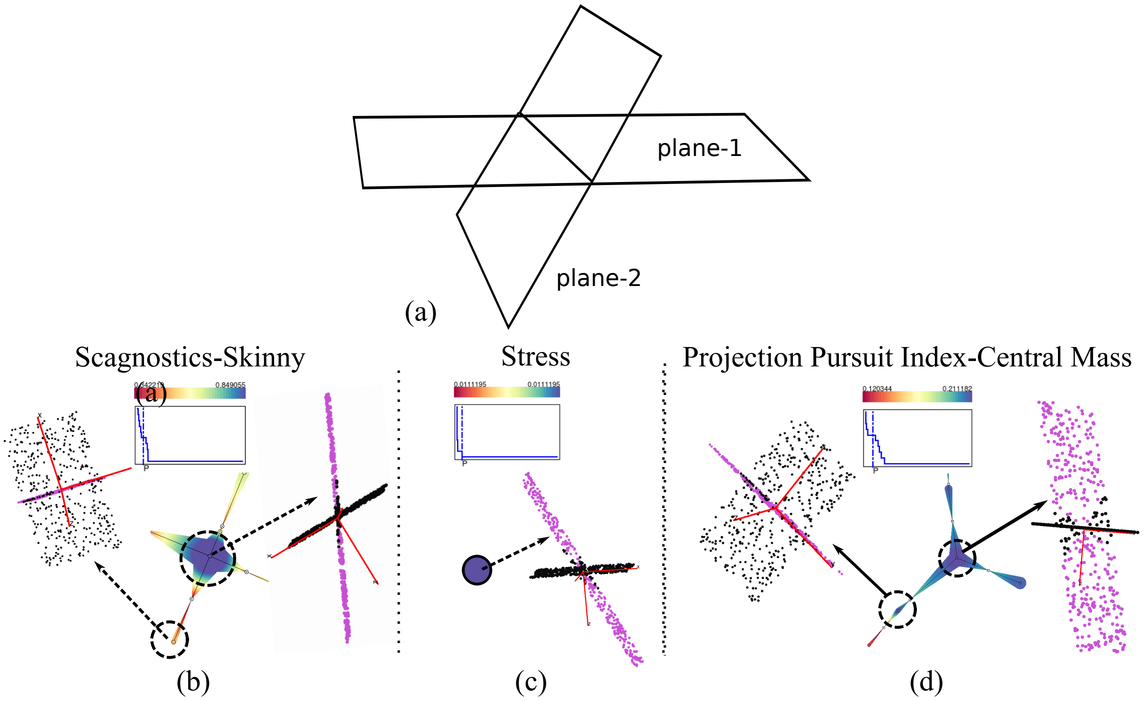


Figure 7.7: Validate the *Grassmannian Atlas* framework on a synthetic two-planes dataset. The dataset is sampled from the space illustrated in (a). In (b), the two maxima within the topological spine correspond to the projections where one or both planes are at the "skinniest". In (c), the (global) stress measure captures only one interesting projection at its global maxima. In (d), the projection pursuit index *central mass* measure captures two projections where one of the two planes becomes "skinny".

7.6.1 Word Embedding Dataset

The following study of Word2Vec dataset is a collaboration with an expert in natural language processing (NLP). The popular Word2Vec algorithm [129] learns a vector space representation of words by modeling the intrinsic semantics of large text corpora. It consolidates the statistical relationships between words in an abstract high-dimensional feature space. According to my collaborator, the analysis and visualization approach for such a dataset is very limited. Often, the t-SNE [59] nonlinear projection algorithm is used for visualization, but most relationships in Word2Vec are linear in nature. He suggests a visualization tool that can produce interesting linear projections to emphasize semantic properties in different parts of the data could lead to valuable new insights.

The complete Word2Vec dataset is obtained by running the Word2Vec algorithm on corpora of news articles, containing 100 billion words. The dimension of the resulting vector representations for the words is fixed at 300. The data used in the experiment is a small subset of the Word2Vec dataset, containing 900 frequently occurring words obtained from the Google analogy task list. This list contains pairs of words with a semantic or syntactic relationship between them, e.g., (queen, king) and (man, woman). Following this, I use PCA to reduce the dimension of the word vectors to 5D in order to reduce the sampling cost. Note that subsampling and dimension reduction are both common strategies in NLP to limit the complexity of the input data without introducing significant errors. To provide a context for the visualization, the 900 words are labeled with 10 categories such as adjective, adverb, verb, and different groups of nouns (e.g., capitals and countries in different continents, states of the US, etc.).

As shown in the quality measure comparison analysis in Section 7.7, several measures, such as *clumpy*, *outlying*, which are more likely to identify multiple complementary projections. In addition, *clumpy* by definition will likely highlight cluster-like features. As demonstrated in Figure 7.8, the *clumpy* measure helps capture the projections that reveals interesting semantic relations in the analogy dataset. The largest maxima (shown on the right) correspond to a projection that clearly separates cities and countries from all other words and does well in separating their respective continents (e.g., orange for North America, dark green for Europe, and blue for South America). A second projection (shown on the left) does less well on cities and countries, but nicely separates the remaining groups of words. My collaborator considers the left projection to be the most informative overall, yet it does not have a very high global ranking, and it would likely be ignored in a ranking-based approach.

A one-on-one session is carried out to obtain meaningful feedback from the collaborator. First, a carefully prepared demo by the researcher is presented to the collaborator. Then the collaborator is directed to experiment with the tool to explore the various measures and projections interactively. The session is concluded by a discussion regarding the capability and usability of the tool. My collaborator shows great interest in the capability of the

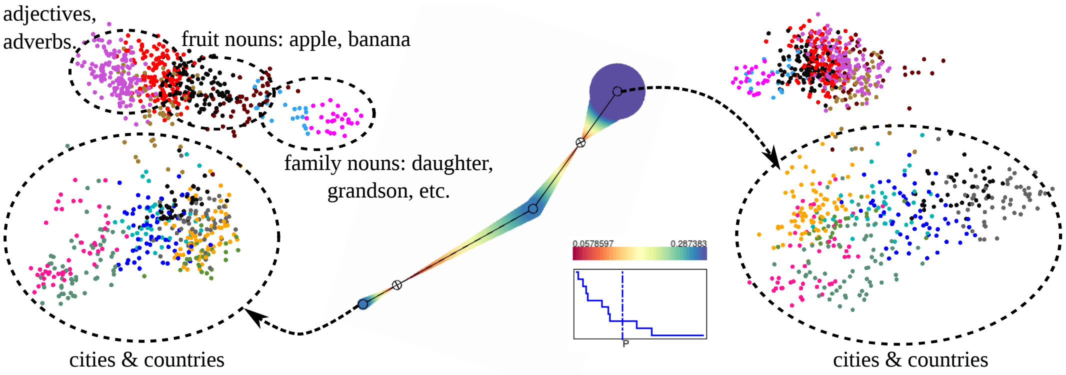


Figure 7.8: Word2Vect dataset. The *clumpy* measure helps to identify the two projections that highlight clear separation between cities and countries from the rest of the data points.

proposed framework. He points out that the Grassmannian Atlas framework can be a useful tool for exploring the word feature space, especially considering it does not have any restriction on what quality measures can be adopted. For example, he suggests new measures specifically tailored towards text analysis can be designed by incorporating semantic relationships among words. Regarding the possible challenges for using the proposed tool, the collaborator points out the basic concept can be challenging to digest at first, since it approaches the problem from a fundamentally different perspective (the space of all linear projections).

7.6.2 E. coli. Dataset

The proposed approach is also applied to biological dataset. As shown in Figure 7.9, based on the *clumpy* quality measure, the framework identifies multiple interesting projections for the E. coli dataset that capture meaningful biological relationships.

The data points (corresponding to different E. coli strains) in the two highlighted projections form clear clusters that are well aligned with the localization site classification labels (see details in [144]). The black corresponds to the *cytoplasm* localization site, which comprises *cytosol* (the gel-like substance enclosed within the cell membrane) and the *organelles* (the cell's internal sub-structures); the purple represents inner membrane without signal sequence; the orange contains inner membrane with uncleavable signal sequence; the light green corresponds to outer membrane; the brown (with only 5 points) is the outer membrane lipoprotein; and the dark green corresponds to *perisplasm*, a concentrated gel-like matrix in the space between the inner cytoplasmic membrane and the bacterial outer

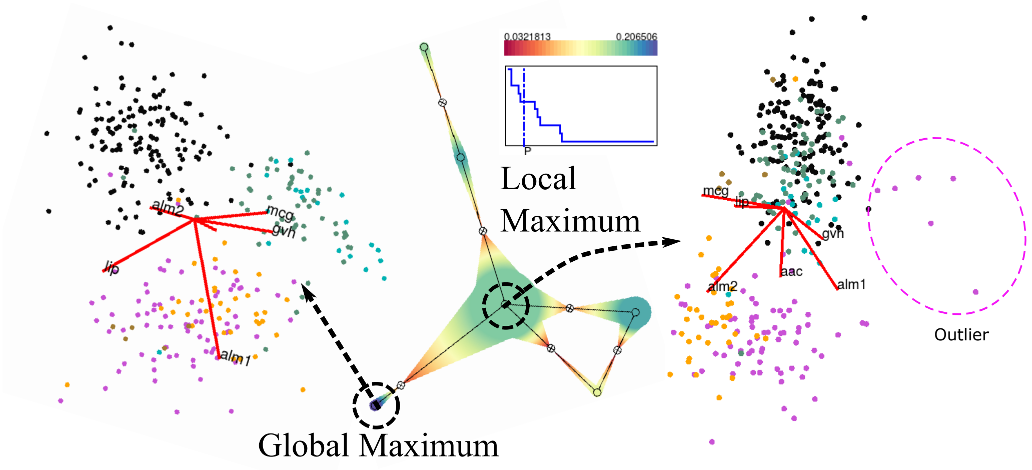


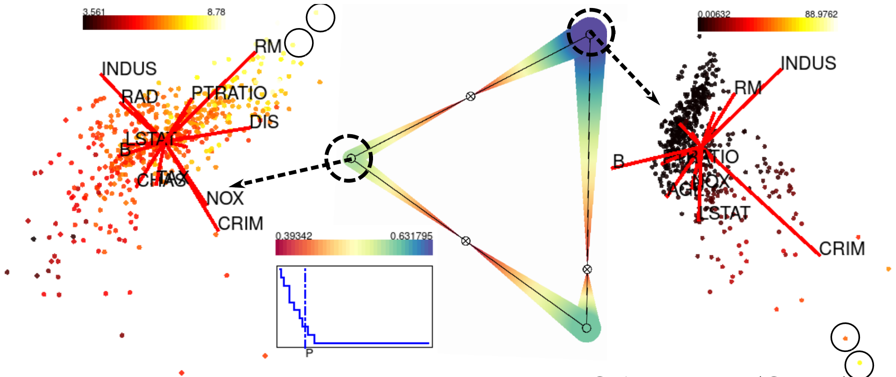
Figure 7.9: The complementary projections captured by *Grassmannian Atlas* using the scagnostics *clumpy* measure for the E. coli dataset.

membrane. The projection at the global maxima captures clear separation between the black, and the (light and dark) green points, separating materials from the inner membrane to the ones from or close to the outer membrane. On the other hand, the projection at the local maxima merges the black with the green points. Both projections group the purple and orange points into one cluster that contains information regarding the inner membrane.

7.6.3 Housing Dataset

In this example (see Figure 7.10), a set of housing data is studied in which each entry records certain property characteristic (14 in total), such as crime rate, median property value, average number of rooms per dwelling, etc. of towns in Boston area. By utilizing the proposed framework and examining the topological spine and corresponding projection computed from the *outlying* measure, I am able to identify some interesting outliers which shed light on the large socioeconomic inequality correlated with the geological separation.

As shown in the projection on the right, I am able to identify outliers that correspond to towns with a comparatively very high crime rate. The difference is so extreme that this outlying pattern is strongest among all the linear projection samples. By looking at one of the local extrema (the projection on the left), we can see the average number of rooms also are correlated with some outliers. After examining the individual data points, the outliers corresponding to the towns that have around 8-9 average rooms per dwelling, while at the same time the minimal number is around 3.5.



Average Number of Rooms (RM) Crime Rate (CRIM)

Figure 7.10: The different outliers captured by the *Grassmannian Atlas* using the scagnostics *outlying* measure for the housing dataset. The outliers are highlighted by small solid circles.

7.7 Quality Measure Comparisons

In this section, I compare the topological structures of various metrics on different datasets to better understand the behavior of each metric.

The Grassmannian Atlas not only helps to identify complementary projections and summarize the structure of quality measures, but also provides an avenue for examining and comparing high-level structures of quality measures in general. In particular, the persistence plot encodes a number of interesting properties in a concise and intuitive manner. As discussed in Section 7.4, the persistence plot records the number of salient local maxima depending on the simplification threshold. In general, the most interesting feature in a persistence plot is the number and width of stairs. Multiple stairs indicate several sets of complementary projections, and the width encodes how stable these features are.

I compute the persistence plots for all 16 quality measures (9 scagnostics, 3 projection pursuit indices, 4 based on objective functions of dimension reduction techniques) and include 11 of these in Figure 7.11. For each measure I evaluate its behavior for five datasets: (i) 2-planes synthetic dataset (3D), (ii) UCI Iris dataset (150 samples in 4D), (iii) UCI E. coli dataset (332 samples in 6D, a subset of the original 336 samples in 8D), (iv) olive oil dataset (572 samples in 8D), and (v) housing dataset (506 samples in 14D). The details for each dataset can be found in the UCI machine learning repository[1].

As shown in Figure 7.11, surprisingly few measures ever show more than two or three complementary projections based on the number of wide stairs in their persistence plots, and the *stress* measure captures a single robust projection in most cases. Such an observation has important implications for ranking-based projection selection - selecting more projections would most likely result in information redundancy.

The significant discrepancies among the topological structures of different quality measures can be explained by their formulations and design goals. The *stress* measure originates from the objective function of MDS [142], and is designed to create a single embedding that best preserves the pairwise distances. Therefore the *stress* measure typically produces a single projection that is optimal on average. On the other hand, quality measures that focus on evaluating the quality of projections based on local structure preservation typically provide multiple, complementary projections. As shown in Figure 7.11, the *clumpy*, *outlying* measures are some of the more effective ones for identifying complementary projections.

In general, given an appropriate quality measure, the Grassmannian Atlas can reliably identify potentially diverse and locally optimal projections. Compared to conventional

[1] http://archive.ics.uci.edu/ml/

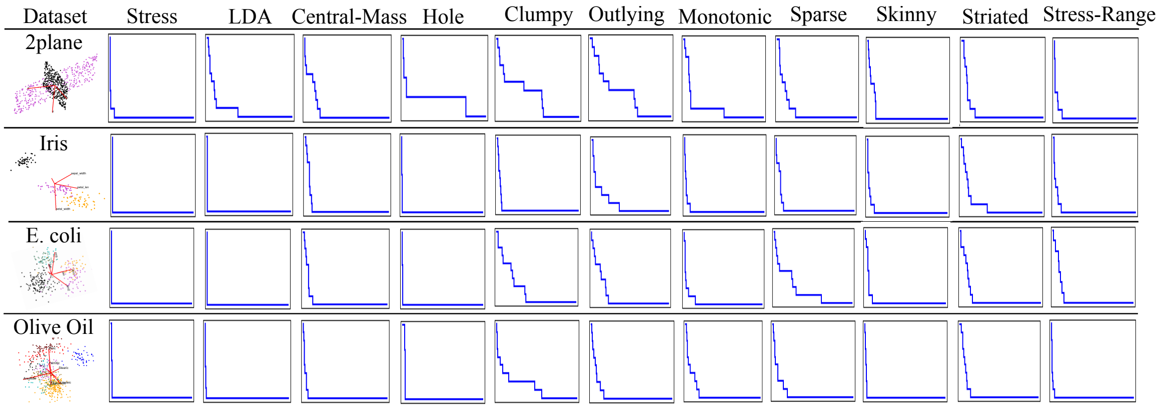


Figure 7.11: Quality measures comparison by evaluating their respective persistence plots, which provide concise summaries of the multiresolution topological structure. Only four datasets are shown here due to space constrains.

rank-based approaches, the proposed framework summarizes the structural relationships among projections according to the topology of the quality measure, and provides a more reliable and locally optimal set of projections for visualization.

PART IV

UNDERSTANDING HIGH-DIMENSIONAL STRUCTURE VIA DATA MANIPULATION IN 2D PROJECTIONS

CHAPTER 8

MEASURING ERRORS IN 2D PROJECTIONS

8.1 The Deceiving Aspect of 2D Projection

The idiom “seeing is believing” was originally used to emphasize the importance of evidence: “only physical or concrete evidence is convincing.” However, in the context of visualization, especially among the end users of a visualization system, “seeing is believing” can be used to convey a common mentality, in which users (mistakenly) believe the visualization they see directly corresponds to the truthful underlying data. The deceiving aspect of visualization is particularly relevant when the inability to “see” the high-dimensional space directly is combined with the unavoidable information loss in generating a 2D projection of high-dimensional data.

Take t-SNE [59] for example, which is a widely used nonlinear dimensionality reduction (DR) method. According to a domain expert in the field of machine learning, users of t-SNE often interpret the inconsistency (e.g., a point that does not seem to belong to its 2D neighborhood) in the projection as the noise in the data, before even considering the inaccuracies in the visualization. However, the misplaced points are very likely introduced by the visualization due to the inability of a 2D projection to faithfully express complex high-dimensional relationships. Often, the notion of “seeing is believing” is so ingrained in our subconsciousness that even expert users, who are well-aware the information loss during the dimension reduction process, need extra help to correctly interpret information in visualization.

8.2 Evaluate 2D Projection Through Distortion Measures

To address the problem of misleading information in a 2D projection, effectively conveying the inaccuracies is essential. The inaccuracies in the projection can be evaluated from two perspectives: first, a global measure of the absolute magnitude of the error, which addresses the question: Is the projection totally misleading? Second is a per-point estimation of the error, which addresses the question: Should I trust a given point in the projection? For the t-SNE example discussed previously, a per-point estimation of the error will provide the user with adequate information to determine whether the inconsistency in the projection is likely to have been introduced by the dimension reduction process.

The concept of quality assessment for projection is not new. Various quality measures of dimensionality reduction (DR) have been proposed, primarily in the machine learning community, for both labeled and unlabeled data. In this dissertation, these measures are referred to as *distortion measures.* For labeled data, distortion measures that focus on classification errors [145] or group memberships [146] seem to be obvious choices. For instance, the quality of group compactness [146] measures consistency among group memberships in the local neighborhood of a point, based on labeled information. For unlabeled data, some criteria for evaluation relate pairwise distances through a direct comparison between high- and low-dimensional space. For example, quality of distance mapping [146] computes the correlation coefficient between the pairwise distance matrices before and after DR. Measurements such as *strain* [147] and *stress* [142] (described in Section 8.3) capture absolute differences between distance matrices. Other criteria do not directly compare lengths but rather ranks of pairwise distances. Criteria such as *precision and recall* [148], *co-ranking* [67], *quality of point neighborhood preservation* [146] and *agreement rate* [149] focus on calculating the average number of neighbors that agree in high and low dimensions. Such rank-based criteria are typically scale-independent in the sense that they are invariant under linear transformations of distances. Specific measurements of geometrical and topological distortions, due to manifold compression, stretching, gluing and tearing, have been proposed and visualized in [150].

In this dissertation, the concept of distortion measure is extended and refined to classify them into two categories [31]: these general measures that are applicable to various types of projections, as well as the DR-specific measures that are applicable only to the projections generated by a specific DR method. As pointed out by Lee et al. [67], a natural way to assess the quality of DR is to look at the value of the objective function after optimization. This idea is adopted in this dissertation for deriving the pointwise (local) distortion measures from formalized objectives of DR methods. In addition, this research also introduced two new distortion measures based on robust distance and kernel density estimate. In the next section, a systematic discussion of the global and pointwise distortion measures is presented.

8.3 Pointwise Distortion Measures

Pointwise (local) distortion measures provide the foundations for visualizing the inaccuracies in 2D projections. In this section, a systematic overview of global and pointwise distortion measures for several popular DR techniques is given. The first type of distortion measures quantifies the cost on structural transformation from high-dimensional to low-dimensional spaces. It is derived from the particular objective function a given DR technique is formulated to optimize; thus it is DR-dependent, as described in Section 8.3.1. The second type of distortion measures is DR-independent and focuses on computing distance distortions, density differences or ranking discrepancies[68], applicable across DR techniques, as described in Section 8.3.2.

The basic setting for DR is as follows: given a set of n points $X = \{x_1, x_2, ..., x_n\}$ in $\mathbb{R}^l$, find a set of points $Y = \{y_1, ..., y_n\}$ in $\mathbb{R}^m$ where $m \ll l$, such that Y represents X by preserving certain structural properties of X. For visualization purpose, $m = 2$, with possible extension to $m = 3$. For a given DR technique, a global distortion measure assigns a real-valued number to the pair (X, Y), which gives an overall, coarse quality assessment, whereas a pointwise distortion measure is a function that maps points in X to $\mathbb{R}$, which provides localized, fine quality assessment.

8.3.1 DR-Dependent Distortion Measures

Most DR techniques can be formulated as optimization problems formalized with objectives. For the popular DR techniques described below, optimizing the objectives is typically formulated as minimizing certain cost functions. A cost function incorporates a natural quality measure that assesses how much structure, in terms of relations among data points in high dimensions, stays consistent with the one inferred by the low-dimensional embedding; or alternatively, how much cost is needed in transforming one to another. Such a cost function gives rise to a natural global distortion measure $\mathcal{E}$ to assess the overall quality of the DR, and its pointwise derivation leads to a local distortion measure $\varepsilon : X \to \mathbb{R}$ that captures how much a point contributes to the global distortion and how well it agrees with its neighbors. Finally, the following relationship is enforced.

$$\mathcal{E} = \sum_i \varepsilon(x_i)$$

Principle Component Analysis. PCA finds the directions of projection such that the squared distance of the points to these directions is minimized. Let $\mu : \mathbb{R}^l \to \mathbb{R}^l$ be a certain projection map. PCA seeks to minimize the global cost over μ, $\mathcal{E} = \sum_i ||x_i - \mu(x_i)||^2$, and the corresponding local cost ε is defined as, $\varepsilon(x_i) = ||x_i - \mu(x_i)||^2$.

The map μ is defined by the orthogonal direction with respect to a hyperplane defined by a collection of orthogonal basis $\{u_1, u_2, ..., u_m\}$ (where $u_i \cdot u_i = 1$ and $u_i \cdot u_j = 0$ for $i \neq j$). The projection $\hat{x_i} := \mu(x_i) \in \mathbb{R}^l$ of a given point $x_i \in X$ under μ could be written as $\hat{x_i} = \bar{x} + \sum_{j=1}^{m} z_j^i u_j$, where the mean $\bar{x} = \frac{1}{m}\sum_i x_i$, and $z_j^i = (x_i - \bar{x}) \cdot u_j$. Now the global cost can be written as:

$$\mathcal{E} = \sum_i ||x_i - \hat{x_i}||^2$$

and the local cost:

$$\varepsilon\,(x_i) = ||x_i - \hat{x_i}||^2$$

Classic Multidimensional Scaling. MDS is commonly referred to as a class of techniques rather than a specific algorithm. cMDS [147], also known as Principle Coordinate Analysis (PCoA) or Torgerson Scaling, is closely related to PCA. In cMDS, the distance is converted to inner production dissimilarity and *strain* is optimized though an Eigenvalue decomposition.

Let b_{ij} be the inner product between a pair of points x_i, x_j in $\mathbb{R}^l$ and $\hat{b}_{ij}$ be the corresponding inner product in $\mathbb{R}^m$. That is, treating points as vectors, $b_{ij} = x_i \cdot x_j$ and $\hat{b}_{ij} = y_i \cdot y_j$. The relationship between distance matrix and inner product matrix can be defined as, $d_{ij}^2 = b_{ii} - 2b_{ij} + b_{jj}$, where d_{ij} corresponds to the Euclidean distance between x_i and x_j . The global cost is defined to be equal to the strain, that is,

$$\mathcal{E} = \frac{\sum_{i,j}(b_{ij} - \hat{b}_{ij})^2}{\sum_{i,j} b_{ij}^2}$$

The local cost corresponds to the pointwise strain,

$$\varepsilon\,(x_i) = \frac{\sum_{j}(b_{ij} - \hat{b}_{ij})^2}{\sum_{i,j} b_{ij}^2}$$

Laplacian Eigenmap. The Laplacian Eigenmap [54] (LE) algorithm proceeds by first constructing an adjacency graph on X based on either k-nearest neighbor (KNN) graph or ϵ-neighborhood. If x_i and x_j are connected by an edge, the weight w_{ij} is either defined as a heat kernel, that is, $w_{ij} = \exp(-||x_i - x_j||^2/t)$ (with diffusion parameter t), or simply defined as $w_{ij} = 1$; otherwise $w_{ij} = 0$. LE seeks to minimize a global cost function,

$$\mathcal{E} = \sum_{i,j} ||y_i - y_j||^2 w_{ij}$$

Under appropriate constraints. The corresponding local cost is:

$$\varepsilon\,(x_i) = \frac{1}{2}\sum_j ||y_i - y_j||^2 w_{ij}$$

Isomap. Isomap[52] is a nonlinear DR technique based on cMDS. In Isomap, the distance between pairs of points is geodesic distances approximated by the shortest paths between pairs of points in a neighborhood graph. Therefore the cost function is the same as cMDS except the Euclidean distance matrix is replaced by an approximated geodesic distance matrix.

Locally Linear Embedding. LLE [53] represents each point (in $\mathbb{R}^l$) as a weighted linear combination of its neighbors and tries to preserve this linear relationship in the reduced dimension $\mathbb{R}^m$. It optimizes the following global cost,

$$\mathcal{E} = \sum_i ||y_i - \sum_j W_{ij} y_j||^2$$

where W_{ij} is the weight matrix that stores such a linear relationship. The local cost can be written as,

$$\varepsilon\,(y_i) = ||y_i - \sum_j W_{ij} y_j||^2$$

8.3.2 DR-Independent Distortion Measures

DR-independent criteria, on the other hand, can be applicable to a collection of DR techniques, and are inspired by measurements of distance distortions, density differences or ranking discrepancies. Some nonlinear DR techniques, such as LE, use constraints in their algorithms to remove an arbitrary scsaling factor in the embedding. Points in the reduced dimension are therefore computed under a fixed scale, which means that ranges of values in $\mathbb{R}^l$ and $\mathbb{R}^m$ differ drastically, rendering the scale-dependent distortion measures such as local stress, robust distance distortion and kernel density estimate distortion meaningless. To address this issue, two types of scaling factors is used. The first one computes the ratio between the radiuses of minimum enclosing balls [151] of the data in $\mathbb{R}^l$ and $\mathbb{R}^m$ to rescale the embedding. The second type, which is also less sensitive to outliers, computes the ratio of average distances to the centroid.

Kernel Density Estimate distortion. Here a novel class of distortion measures based on a kernel density estimate (KDE) is introduced. Each of these measures (based on a chosen kernel) quantifies differences in densities among local neighborhoods. In addition, a multiscale version of the measure is easily attainable by varying the parameters associated with a given kernel; thus it allows adaptive data explorations. A kernel is a non-negative similarity measure $K : \mathbb{R}^l \times \mathbb{R}^l \to \mathbb{R}^+$ where more similar points have higher value. Gaussian kernel is considered here, where $K(p, x) = \exp(-||p-x||^2/2\sigma^2)$. A KDE is a way to estimate a continuous distribution function over $\mathbb{R}^l$ for a finite point set $P \subset \mathbb{R}^l$. Specifically,

$$KDE_{\,P}(x) = \frac{1}{|P|} \sum_{p \in P} K(p, x)$$

The distortion function measures differences between KDE in $\mathbb{R}^l$ and KDE in $\mathbb{R}^m$. That is, the global KDE distortion,

$$\mathcal{K} = \sum_i |KDE_X(x_i) - KDE_Y(y_i)|$$

and the local KDE distortion,

$$k\,(x_i) = |KDE_X(x_i) - KDE_Y(y_i)|$$

Stress. This distortion measure is based upon an objective function used in a distance scaling version of MDS, referred to as *stress*. The stress is used to measure distance distortions. Let d_{ij} be the distance between a pair of points i, j in $\mathbb{R}^l$ and $\hat{d}_{ij}$ be the corresponding distance in $\mathbb{R}^m$. Global stress is defined as,

$$\mathcal{S} = \frac{\sum_{i,j}(d_{ij} - \hat{d}_{ij})^2}{\sum_{i,j} d_{ij}^2}$$

Local stress is,

$$s\,(x_i) = \frac{1}{2} \cdot \frac{\sum_j (d_{ij} - \hat{d}_{ij})^2}{\sum_{i,j} d_{ij}^2}$$

Robust distance distortion. A distortion measure inspired by robust MDS (rMDS) [152, 153] is introduced. It shares similarities with stress but is proved to be more robust with respect to noise and outliers. The global robust distance distortion is defined as,

$$\mathcal{R} = \frac{\sum_{i,j} |d_{ij} - \hat{d}_{ij}|}{\sum_{i,j} |d_{ij}|}$$

The local robust distance distortion is,

$$r\,(x_i) = \frac{\sum_j |d_{ij} - \hat{d}_{ij}|}{\sum_{i,j} |d_{ij}|}$$

Co-ranking distortion. Despite not being the contribution of this research, in the software system a rank-based, scale-independent criterion derived from co-ranking matrices [67, 68] is included for completeness. Let d_{ij} be the distance between a pair of points x_i, x_j in $\mathbb{R}^l$ and $\hat{d}_{ij}$ be the corresponding distance between y_i, y_j in $\mathbb{R}^m$. The *rank* of x_j with respect to x_i is

$$\rho_{ij} = |\{k \mid d_{ik} \le d_{ij} \quad or \quad (d_{ik} = d_{ij} \quad and \quad 1 \le k < j \le N)\}|$$

Similarly, the rank of y_j with respect to y_i is

$$\gamma_{ij} = |\{k \mid \hat{d}_{ik} \le \hat{d}_{ij} \quad or \quad (\hat{d}_{ik} = \hat{d}_{ij} \quad and \quad 1 \le k < j \le N)\}|$$

where $|\cdot|$ denotes set cardinality. The difference $R_{ij} = r_{ij} - \rho_{ij}$ is considered *rank errors.* The co-ranking matrix C is defined by

$$C_{kl} = |\{(i,j) \mid \rho_{ij} = k \quad and \quad \gamma_{ij} = l\}|$$

A DR with no errors would produce a diagonal co-ranking matrix.

In [67], a quality for dimension reduction is proposed as a sum of partial entries in the co-ranking matrix,

$$Q = \frac{1}{Kn} \sum_{k=1}^{K} \sum_{l=1}^{K} C_{kl}$$

where K corresponds to the number of neighbors under consideration. Therefore every co-ranking matrix C can be decomposed into a per-point permutation matrix C^i for every point x_i, with $C = \sum_{i=1}^{N} C^i$ and

$$C^i_{kl} = |\{j \mid \rho_{ij} = k \quad and \quad r_{ij} = l\}|$$

The pointwise contributions is

$$Q_i = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=1}^{K} C^i_{kl}$$

where $Q = (\sum_{i=1}^{N} Q_i)/N$. For a given point, a larger Q_i corresponds to less local distortion. Therefore, the global co-ranking distortion is defined as $\mathcal{Q} = -Q$ and local co-ranking distortion as $q = -Q_i$.

CHAPTER 9

DISTORTION-GUIDED STRUCTURE-DRIVEN MANIPULATION

9.1 Interpret the Errors Via Structure-Driven Manipulation of 2D Projections

For a given projection, the per-point distortion measures address the question of where the inaccurate areas are. However, relying on the distortion measure alone, we still cannot answer the question of why some of the highly distorted areas exist (i.e., why most errors occur in a particular area). Also, ultimately, how do we obtain insights regarding the structures of the data via explorations of their 2D projections (linear and nonlinear)? In this research, an interactive visualization framework [31], in which the distortion measures under dynamic setting are used as the feedback during exploration, is introduced to address these challenges.

Visualizing pointwise distortions under the static setting illustrates the qualitative disparities among different regions of the embedding, which in turn, reflect structural discrepancies within the original data. Regions with higher distortions correspond to areas with more structural uncertainty (and equivalently, less structural preservation). When examining the cause of higher pointwise distortions, we may ask whether the existence of such distortions are due to the high-dimensional structures of the original data that are hidden in its embedding? Furthermore, is it possible for the user to manipulate the locations of some points in the embedding in order to achieve better pointwise distortions locally, and what would such a manipulation tell us about the original data? These questions motivate the proposed research to compute and visualize distortion measures under a dynamic setting, where on-the-fly updates of pointwise distortions due to data movement and data deletion reflect structural relations among different parts of the data. Such data manipulations in the visual space do not trigger a new DR optimization process, but result in updates of relevant distortion measures, which offer valuable feedback as to how much the manipulated results deviate from the original embedding. By moving subsets of points, an increase (or decrease) in distortion measures indicates structural dependencies (or independencies, respectively) among different parts of the data, which may lead to new and valuable insights.

One fundamental challenge when manipulating projected data points in 2D is the lack of high-dimensional structure information. Due to the constraints and limitations of 2D space, common interaction tools such as lasso or box selection may select points that belong to faraway high-dimensional neighborhoods, which introduce more inaccuracies rather than helping resolve structural ambiguity. Meaningful data manipulations (e.g., data movement and data deletion) in the visual space should be structure-driven, that is, the selected points should respect certain structures of the original high-dimensional data. In order to effectively manipulate high-dimensional structure in 2D, a skeleton is imposed onto the embeddings computed from hierarchical clustering results, which serves as structural abstractions of the data at multiple scales. The proposed framework allows users to choose from two classes of built-in clustering methods: classical (e.g., single- or average-linkage) hierarchical clustering [154] and topological hierarchical clustering based on Morse-Smale complexes [155]. In addition, users can also directly import existing hierarchical clustering results or any nonhierarchical class labeling of the data using a simple file format. These clusters allow users to navigate and manipulate subsets of the data that belong to the same high-dimensional neighborhood at an appropriate level of abstraction.

A typical interactive exploration workflow is illustrated in Figure 9.1, where the key steps are indicated by (a)-(f). (a) Start the exploration by applying a DR technique to the

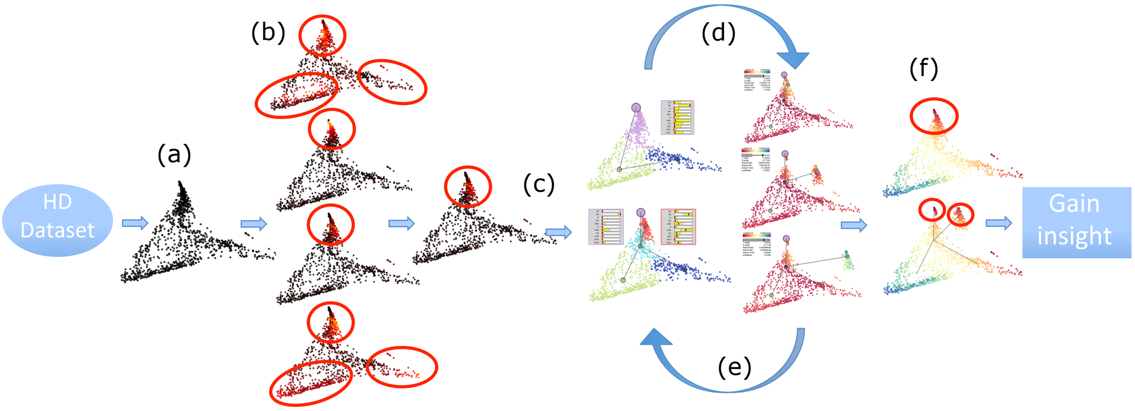


Figure 9.1: A typical interactive data exploration workflow. (a) Dimensionality reduction or projection-finding result; (b) Distortion-guided selection of region of interest; (c)-(d) Hierarchical clustering of the data and distortion-guided clustering selection. (e) Data manipulations with on-the-fly update of distortion measures reveal structural insights of the data. (f) Parameter differentiations across different clusters for additional structural insights.

high-dimensional dataset to obtain the initial projection or obtain the projection from other projection-finding methods (such as the one discussed in PART II of the dissertation). The global distortion measures such as co-ranking could be employed to select a suitable DR and its optimal parameter setting. (b) By visualizing pointwise distortions on the embedding, regions with high distortions across multiple measures (for example) are identified as regions of interest for further investigation. (c) Apply hierarchical clustering of the data to extract the skeleton for manipulation. (d) Use pointwise distortions to guide the clustering selection, where the appropriate level of clustering is chosen based on its agreement with the region of interest. (e) Move and/or delete a subset of data that belong to a targeted cluster in the visual space, where on-the-fly updates of pointwise distortion measures reflect structural relations among different parts of the data. A decrease/increase in distortion measure of the targeted cluster typically indicates structural independencies/dependencies among the target and its neighboring clusters. (f) Obtained further insights regarding differentiating factors among different regions of the data by viewing detailed parameter summary across each cluster

9.2 User Interface and Interaction

In this section, the user interface, user interaction design, and system implementation of the proposed framework are discussed.

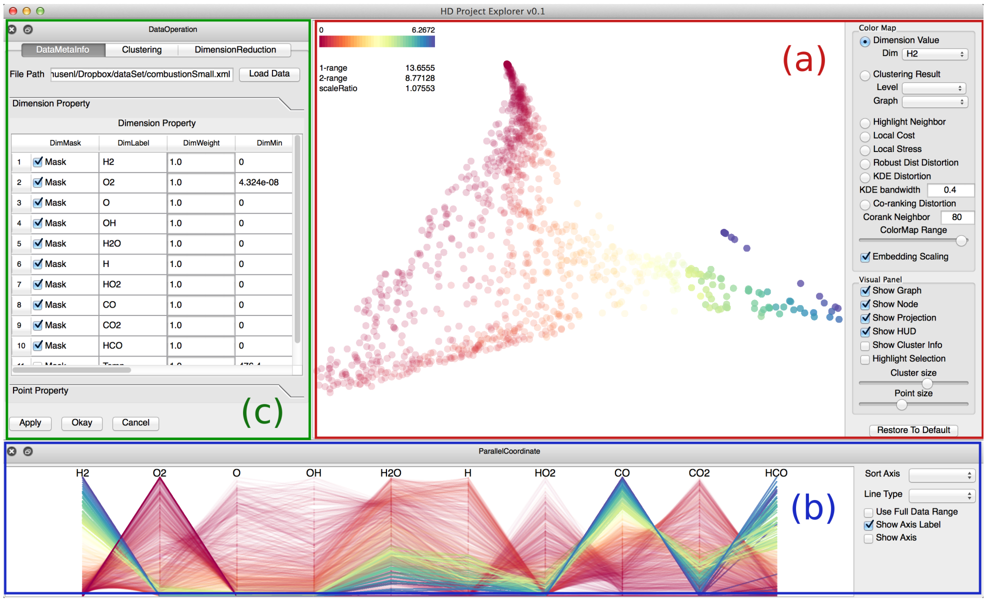
9.2.1 Interface Design

A system overview is shown in Figure 9.2. The overall interface consists of two views and one data operation panel. These visual components are coordinated to provide a more comprehensive visualization of the data. They are interconnected such that selections and changes made in one component will be reflected in others. The system is highly modular and is easily extendable to include additional visual components.

Embedding view. This view is the main canvas of the interface where the results of DR, points embedded in 2D, are visualized. It contains a rich set of user interactions for data exploration. One could apply different colormaps to visualize points by values of a particular dimension, clustering labels or pointwise distortion measures.

Parallel coordinate view. This view displays the original data with each of its dimensions as a vertical axis and each point as a line drawing through each of the axes. A normalization of the range for each axis is optional to suit different usage scenarios.

Data operation panel. This panel contains various data operations such as DR and clustering. The panel is part of the interlinked system so that changes made to the dataset



File Path husenl/Dropbox/dataSet/combustionSmall.xml

Dimension Property

	DimMask	DimLabel	DimWeight	DimMin
1	☑ Mask	H2	1.0	0
2	☑ Mask	O2	1.0	4.324e-08
3	☑ Mask	O	1.0	0
4	☑ Mask	OH	1.0	0
5	☑ Mask	H2O	1.0	0
6	☑ Mask	H	1.0	0
7	☑ Mask	HO2	1.0	0
8	☑ Mask	CO	1.0	0
9	☑ Mask	CO2	1.0	0
10	☑ Mask	HCO	1.0	0

1-range 13.6555
2-range 8.77128
scaleRatio 1.07553

● Dimension Value
Dim H2
○ Clustering Result
Level
Graph
○ Highlight Neighbor
○ Local Cost
○ Local Stress
○ Robust Dist Distortion
○ KDE Distortion
KDE bandwidth 0.4
○ Co-ranking Distortion
Corank Neighbor 80
☑ Embedding Scaling

☑ Show Graph
☑ Show Node
☑ Show Projection
☑ Show HUD
☐ Show Cluster Info
☐ Highlight Selection

Sort Axis
Line Type
☐ Use Full Data Range
☑ Show Axis Label
☐ Show Axis

Figure 9.2: A system overview showing two views and one control panel. (a) Embedding view. (b) Parallel coordinates view. (c) Data operation panel.

are instantly reflected through other views. The panel consists of three sub-panels. The meta-information panel gives a direct view of the data, in terms of its dimensions and statistics, and includes the ability to filter (hide) certain dimensions for analysis; the clustering panel allows the user to select distance metrics, data standardization schemes (e.g., variance normalization) and hierarchical (e.g., classical single-, average-linkage, topology-based) clustering methods, while also allowing loading of existing clustering; and the DR panel enables the user to choose DR techniques and specify their parameters.

9.2.2 Interaction Design

The fundamental principle behind the interaction design is to obtain fresh insights regarding the structure of the data via distortion-guided, structure-driven, interactive manipulations. We provide a list of interaction semantics in the embedding view to aid the manipulations and explorations.

View interactions. Interactions in this category do not cause re-calculation of distortion measures. Typical operations include, point selection through the Lasso tool or cluster-level selection; view zooming and panning; filtering of data points; and selection highlighting. We provide visual aids for the exploration and manipulation operations. In the embedding view, a solid circle (node) represents each cluster center (see Figure 9.3), whose radius scales with the size of the cluster; the nearby cluster centers are connected by gray edges based on the k-nearest neighbor information. These nodes and edges form an abstract skeleton

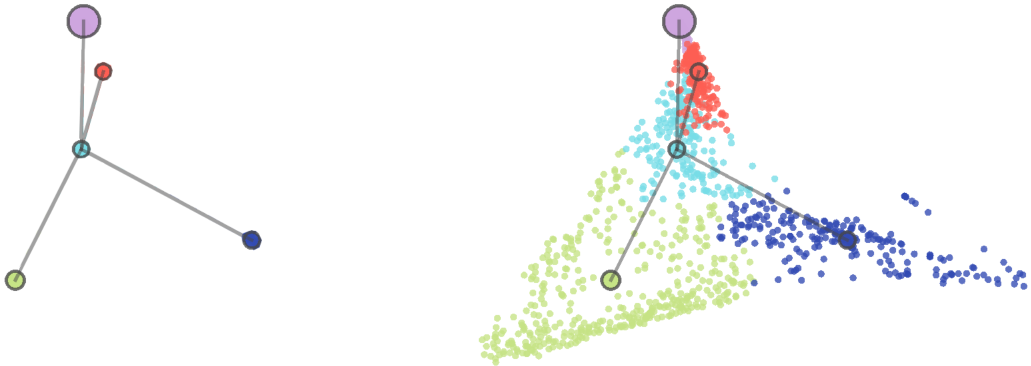


Figure 9.3: Hierarchy skeleton computed from hierarchical clustering is used as a structure-aware handle for manipulating high-dimensional data in 2D.

of the high-dimensional data, which is used as a structure-aware handle for manipulating high-dimensional data. Now, let us take a look at the key exploration operations provided in the system. *Cluster selection* allows the user to select points in a cluster in the view through selection of the cluster center. *Cluster expansion* enables the user to expand a selected cluster on-the-fly to reveal its child clusters. *Cluster compression* merges selected child clusters into their shared parent cluster. A neighborhood graph could also be constructed connecting cluster centers based on their distance proximities, which functions as a structural skeleton.

Data interactions. To visually assist the user to obtain new insights, a set of data manipulations operators (data movement, data deletion) are introduced that cause re-computation of distortion measures. *Data movement* changes the location of selected points via mouse movement. Upon releasing the mouse, both global and pointwise distortion measures are re-calculated and visualized. The increase or decrease of global distortion measure informs the user of the amount of global structural change, while on-the-fly updates of pointwise distortion measures provide valuable information to users regarding structural relations among different parts of the data. *Data deletion* allows users to remove points from the dataset and re-run DR and clustering. Data deletion can remove outliers affecting the DR quality, points with high/low distortions, or hidden/occluded clusters and allow focused analysis of subsets of the data.

9.2.3 Implementation

This distortion guided manipulation is part of an easily extensible software framework. Qt is used for general GUI design and drawing functionalities in views. For DR, an open source C++ library named Tapkee [156] is used. This template-based, easily extensible library provides more than a dozen commonly known DR techniques. In this work, this library is modified to incorporate pointwise distortion calculations so they fit seamlessly in the modular design. The topological hierarchical clustering is based on approximated Morse-Smale segmentation [155]. Both clustering and DR modules are based on APIs that are oblivious to the underlying implementation, and as a result the library implementations could be easily updated or replaced.

9.3 Synthetic Dataset Example

Via a synthetic dataset, the basic functionality, namely, distortion-guided clustering selection, data movement and data deletion in combination with an on-the-fly update of pointwise distortion measures is demonstrated.

Here a parabola dataset is used as a proof-of-concept example, which contains trivial structural information that is easily interpretable in the embedding view. Following the exploration pipeline illustrated in Figure 9.1. Step (a)-(c): apply PCA to the data and obtain a 2D embedding colored by KDE distortions (Figure 9.4(b)). Both KDE distortion and local cost (not shown here) identify a central region of interest (enclosed by the red circle) with low distortion. Step (d): pointwise distortion measures is used to guide the clustering selection where a configuration with five clusters can be obtained after cluster expansions (Figure 9.4(b)-(d)). Step (e): the system allows the user to move points that belong to the blue (central) cluster and update the distortion on-the-fly (Figure 9.4(e)-

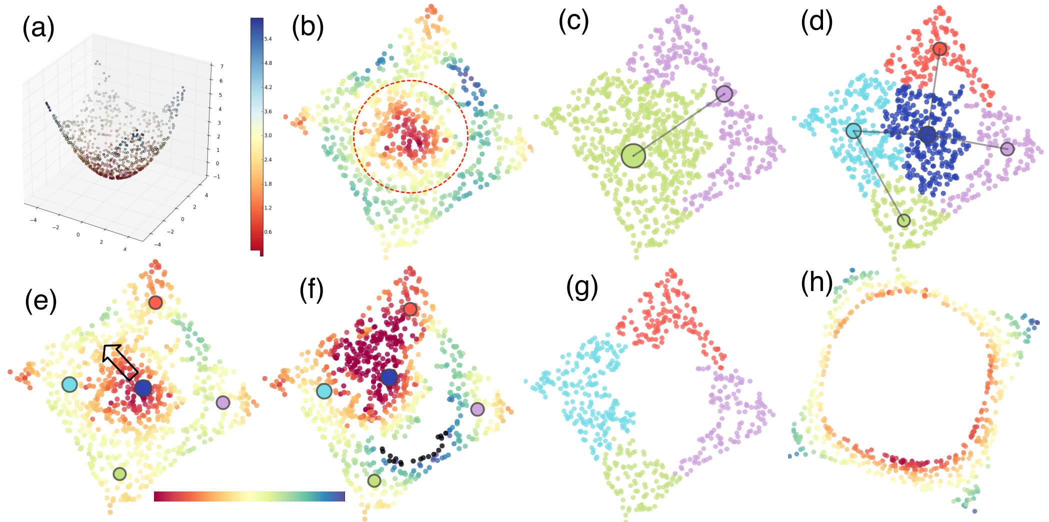


Figure 9.4: Parabola dataset. (a) 3D embedding colored by z-coordinate. (b) 2D embedding colored by KDE distortion. (b)-(d) Distortion-guided clustering selection. On-the-fly update of distortion measures for data movement (e)-(f), and data deletion (g)-(h). Distortion measures adopt spectral colormap.

(f)). A drastic increase in distortion along its boundary indicates a structural dependency among the blue cluster and its neighbors. Finally, after deletion of the blue cluster (Figure 9.4(g)-(h)), DR is re-appled on the remaining points for a more focused study.

9.4 Application Examples

The utility and effectiveness of the proposed framework is showcased through case studies involving real-world datasets from combustion and nuclear simulations.

9.4.1 Combustion Simulation Dataset

This dataset consists of 2.8K samples of chemical composition and temperature extracted pointwise from time-varying jet simulations of turbulent CO/H_2-air flames [120]. The simulation records the concentrations of 10 chemical compounds: H_2, O_2 (Oxygen gas / Oxidizer), O (Oxygen), OH (Hydroxide), H_2O (Water), H (Hydrogen), HO_2, CO (Carbon monoxide), CO_2 (Carbon dioxide) and HCO. The dataset can be modeled as a 10D point cloud with temperatures as observations. The domain scientists are interested in understanding conditions that trigger extinction and re-ignition phenomena, which correspond to points (parameter settings) with minimal temperatures.

The interactive data exploration process follows a typical pipeline illustrated in Figure 9.1. Step (a): Apply cMDS to the dataset, and color the points by temperature. The result is shown in Figure 9.5(a), where two areas are visible with minimal temperatures (marked by arrows), which may correspond to extinction scenarios. Step (b): In order to better understand the DR result and identify the area of interest for further analysis, various pointwise distortion measures are examined (Figure 9.5(b)-(f)). All five of the distortion measures indicate that relatively large distortion exists among points near one of the temperature minima (top area enclosed by the red circle). Such a region becomes the primary target for further investigation.

Steps (c)-(d): Apply classical (average-linked) hierarchical clustering to the data. As illustrated in Figure 9.6(a)-(b), pointwise distortions is used to guide the clustering selection, where the appropriate level of clustering is chosen based on its agreement with the region of interest. Through cluster expansion, a resolution with five clusters (Figure 9.6(b)) is obtained, where the red cluster (pointed by red arrow) agrees well with the region of interest (area enclosed by the red circle in Figure 9.5(b)).

Steps (e): the user moves a subset of the data that belongs to the red cluster away from its neighboring clusters, as illustrated in Figure 9.6(c)-(e). A drastic decrease of pointwise distortion can be observed in the area of interest under moderate movement (Figure 9.6(d)).

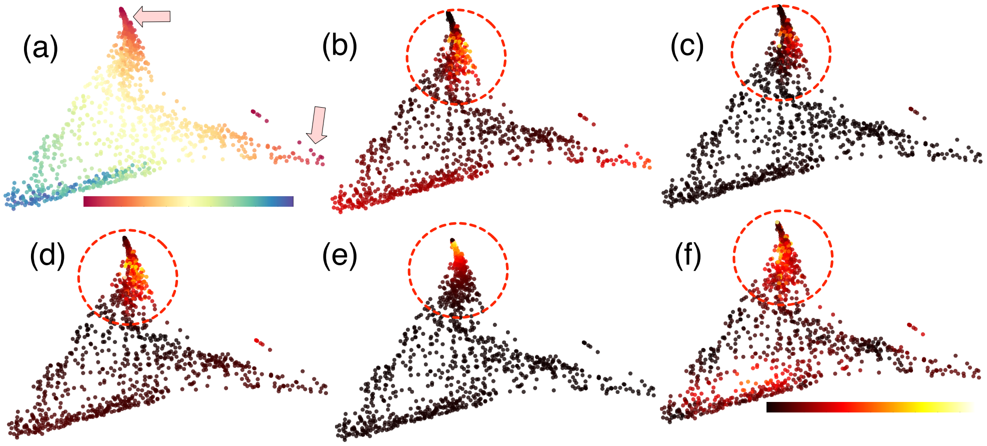


Figure 9.5: Combustion dataset. (a) Points colored by temperature. (b)-(f) All five distortion measures (local cost, local stress, robust distance distortion, KDE distortion and co-rank distortion) indicate an interesting region with high distortion around one of the temperature minima. Temperature image uses the *spectral* colormap and distortion measure images adapt the *hot* colormap.

This indicates a certain level of structural independencies between the red cluster and its neighborhood points. Therefore, the points in the red cluster may potentially correspond to a distinct extinction phenomenon that is different from its nearby cluster. However, further data movement substantially increases the distortion measure (Figure 9.6(e)), which indicates that the red cluster is not completely separated from the rest of the data.

Step (f): To further investigate the nearby red and purple clusters that both contain points with local minimal temperatures, summary statistics of parameters associated with each cluster is display, as illustrated in Figure 9.6(g) (where the red and yellow bars correspond to the mean values and the data range of the labeled parameters). Such summary statistics indicate that the differentiating factor between those two clusters is the vastly different HO_2 concentration (marked by pink arrows). In addition, the proposed tool provides alternative topological hierarchical clustering results to further validate the separation of these local minima, as illustrated in Figure 9.6(f) where the blue cluster (pointed by blue arrow) is a topologically different region (based on the Morse-Smale segmentation) with respect to its neighbors, see [143] for details.

Finally, according to the domain scientists, the red cluster in Figure 9.6(b) represents an independent temperature local minima that correspond to parameter configurations of a special extinction condition (previously unknown to domain scientists as described in [143]), where the mixing of fuel and oxidizer is highly turbulent and blows the flame out, resulting in a large amount of HO_2.

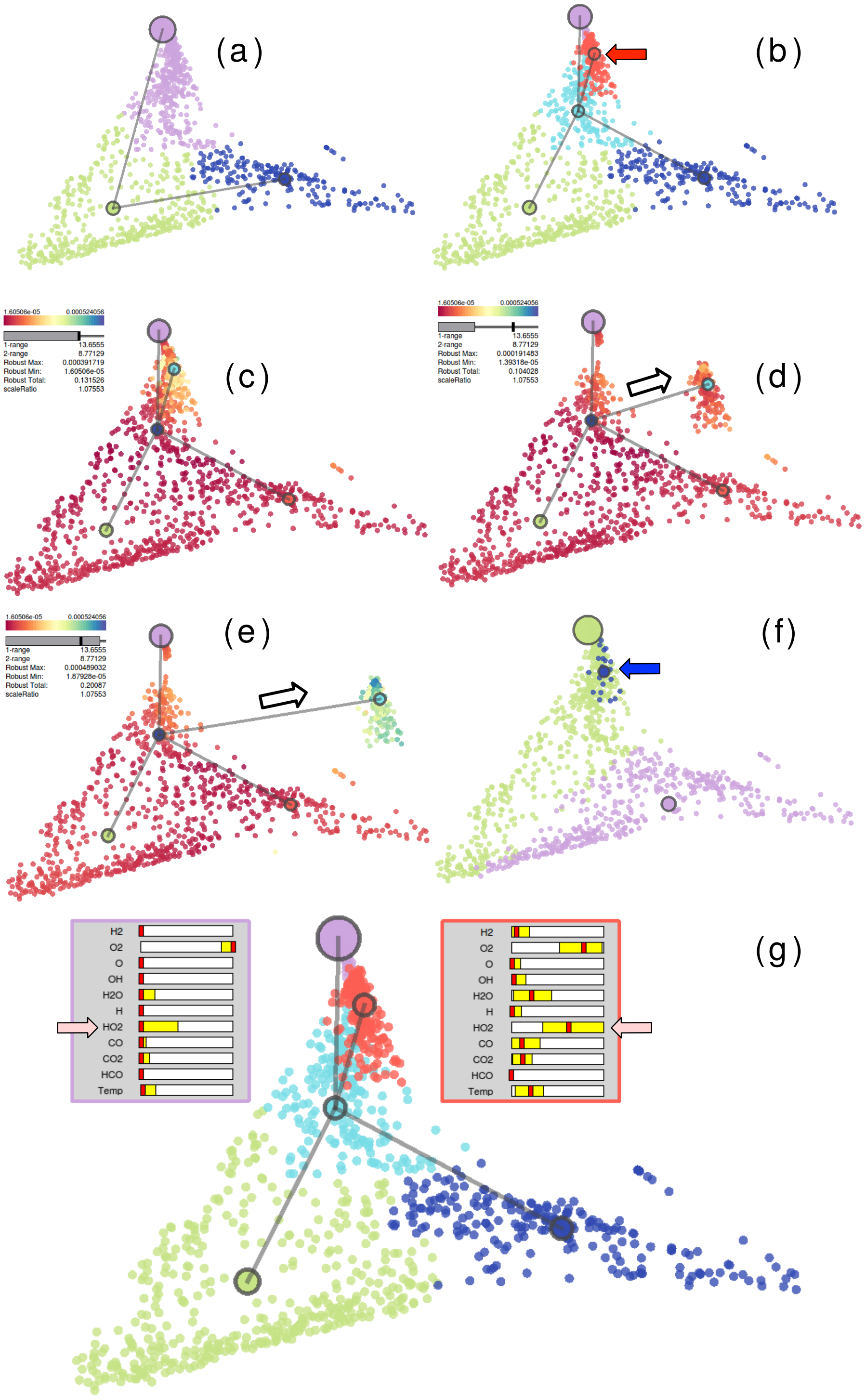


Figure 9.6: Combustion dataset. (a)-(b) Distortion-guided cluster selection. (c)-(e) On-the-fly updates of pointwise distortion measure (local stress) reflect structural relations between different parts of the data. (f) Validation of two overlapped temperature minima based on topological clustering. Distortion is colored by spectral colormap. The parameter boxes in (g) contain summary statistics of parameters in the clusters.

9.4.2 Nuclear Reactor Safety Analysis Dataset

This dataset simulates an accident scenario when a plane crashes into a sodium-cooled fast reactor power plant and destroys three of the four cooling towers [157], and, thus, the reactor core cooling capabilities are disabled. A recovery crew then arrives at the site and attempts to re-establish the cooling of the reactor by restoring the damaged towers one by one, during which time the core temperature keeps increasing if the cooling system is disabled. When the reactor reaches a maximum temperature of 1000K the simulation is considered a system failure scenario; otherwise it is a system success. A set of stochastic parameters, such as crew arrival time and tower recovery time, influence how the core temperature changes over time. An ensemble of 609 transient simulations has been generated, each consisting of a time-varying core temperature profile corresponding to a single simulation. Each profile sampled at 100 time steps and is studied as a 100D dataset. The domain scientists are interested in studying the structure of this dataset and understanding characteristics associated with system failures and system successes, for nuclear reactor safety analysis.

Once again, the analysis is carried out by following the data exploration pipeline (illustrated in Figure 9.1). Step (a): Apply cMDS to obtain a 2D embedding. Step (b): Both local stress and robust distance distortion visualizations (Figure 9.7(a)-(b)) identify an interesting region in the lower part of the embedding (enclosed by the red circle) with relatively high distortions.

Step (c)-(d): Apply classical hierarchical clustering on the data. Through cluster expansion and compression (Figure 9.7(c)), a hierarchical clustering with four clusters where the green cluster agrees almost perfectly with the region of interest is obtained. Step (e): The user then move the points associated with the green cluster away from its neighbors in the visual space, and a small movement increases the distortion measure drastically (Figure 9.7(f)-(g), distortions before and after data movement). This change of distortion indicates that the green cluster is structurally dependent on the rest of the data.

Step (f): Now the embedding with known labels of the data is shown, as illustrated in Figure 9.7(d), where points are colored by their labels of success (purple) or failure (yellow). The green cluster in Figure 9.7(c) agrees almost perfectly with the the yellow cluster (failure cases) in Figure 9.7(d). This offers validation that the distortion-guided clustering selection captures some inherent structure of the data.

By further investigating the local stress and robust distance distortion (Figure 9.7(a)-(b)), there are two points with the highest distortions. These points are marked by arrows in Figure 9.7(a), (b) and (e), where Figure 9.7(e) illustrates all the time-varying core

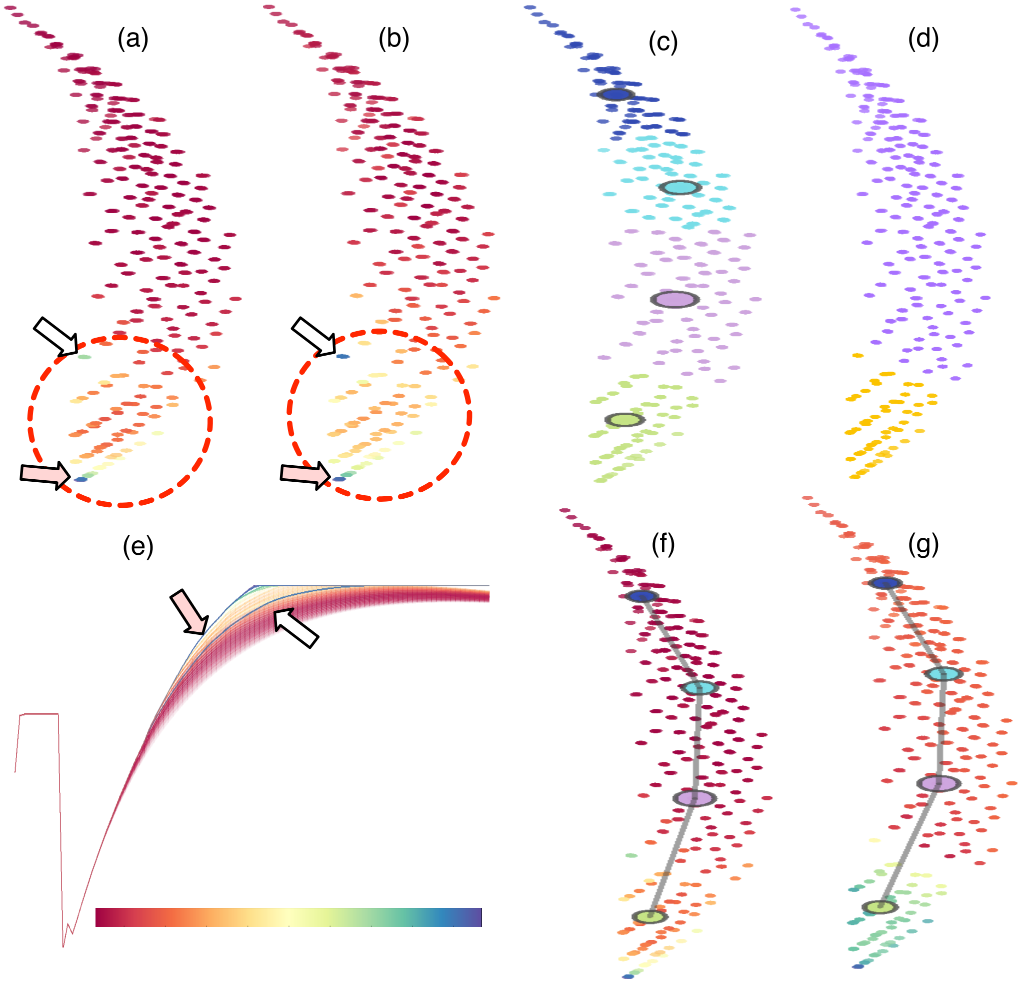


Figure 9.7: Nuclear dataset. (a) Local stress; (b) Robust distance distortion; (c) Distortion-guided cluster selection; (d) Points colored by their labels: system failure (yellow) and system success (purple); (e) Plot of 609 time-varying core temperature profiles in the parallel coordinate plots where x-axis is time, y-axis is temperature. (f)-(g) On-the-fly update of local stress before (f) and after (g) movement of points belonging to the bottom cluster.

temperature profiles in the parallel coordinate plot. The point marked by white arrow corresponds to a boundary scenario that separates system failures from system successes, and the other marked by pink arrow corresponds to a limiting scenario that reaches failure temperature at the earliest simulation time. These distortion-guided observations again offer valuable information of the data.

Furthermore, the analysis can focus on just the system success scenarios by removing all the failure cases. As shown in Figure 9.8(a), all the failure cases are deleted and cMDS is re-applied. Through local distortion visualizations (Figure 9.8(b)-(c)), a point with high

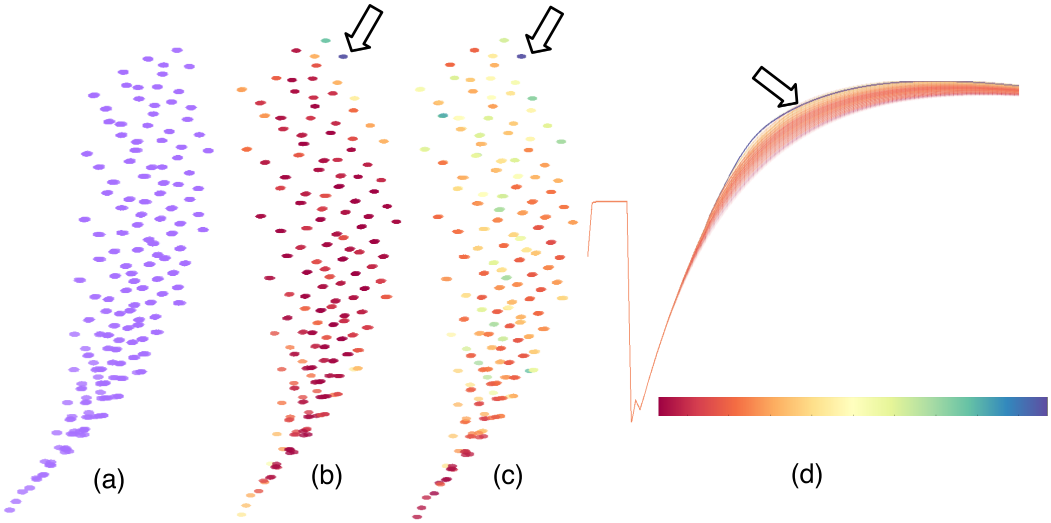


Figure 9.8: Nuclear dataset. (a) Interactive deletion of failure cases; (b)-(c) re-apply DR and visualize by local cost (b) and KDE distortion (c). Both visualizations reveal a point (indicate by white arrow) with high distortion that corresponds to a boundary scenario for the success cases. (d) Success scenarios in parallel coordinate plots.

distortion that corresponds to a boundary scenario can be identifed among the success cases (Figure 9.8(d)).

PART V

CONCLUSIONS AND FUTURE WORK

CHAPTER 10

SUMMARY AND OUTLOOK

10.1 Conclusions

Ever since John W. Tukey popularized the concept of exploratory data analysis [158] and introduced (together with Friedman) the seminal work projection pursuit [9] in the 1970s, understanding high-dimensional space through 2D projections has been regarded as an important and challenging research goal for statisticians and computer scientists alike. In this age of information abundance, multi-parameter datasets have been generated in numerous fields with ever-increasing complexity and size. High-dimensional data visualization techniques are presented with the tremendous opportunity to become one of the standard tools for studying a wide range of applications. Despite many advances in visualization, enormous challenges remain.

This dissertation introduces a visual exploration framework that aims to address some of these visualization challenges. It introduces the subspace analysis approach for identifying 2D projections that reveal intrinsic structures of the dataset (PART II). The subspace analysis approach assumes the high-dimensional dataset can be represented by a mixture of low-dimensional linear subspaces with mixed dimensions, and provides a method to reliably estimate the intrinsic dimension and linear basis of each subspace extracted from the subspace clustering. Subsequently, these bases are used to define unique 2D linear projections as viewpoints from which to visualize the data. To understand the relationships among the different projections and to discover hidden patterns, they are then connected through dynamic projections that create smooth animated transitions between pairs of projections. The view navigation graph, which provides flexible navigation among these projections, is introduced to facilitate an intuitive exploration. This dissertation also proposes an algorithm for generating a structural summary of quality measures in the space of 2D projections (PART III). The Grassmannian Atlas provides a fundamentally unique approach to exploring the space of all linear projections (more specifically linear subspaces), the Grassmannian. By studying quality measures as functions defined on the Grassmannian, users are able to identify local optimal projections as well as obtain an intuitive understanding of the topological structures of these quality measures. The proposed framework not only enables the comparison of multiple quality measures, but also helps to guide the design of and provide benchmarks for new quality measures. Moreover, this dissertation introduces a data manipulation scheme in a 2D projection that aids the understanding of high-dimensional structures (PART IV). The distortion-guided and structure-driven interactive framework facilitates the understanding of high-dimensional data via manipulation of its 2D projections (linear and nonlinear). The structural abstractions obtained through hierarchical clusterings allow multiscale data manipulations, even with hidden or occluded data points in 2D. Pointwise distortion measures are used to guide the cluster expansion and compression process to select the appropriate level of clustering and help users explore meaningful subregions of the data. Combining interactive data manipulations in the 2D projection with on-the-fly updates of distortion measures provides new insights regarding structural relations among different parts of the data. Finally, all the proposed techniques in this dissertation are readily available as components that work together in a self-contained software system, *DataExplorerHD*. To conclude, this dissertation has made meaningful advances that expanded the state-of-the-art.

10.2 Beyond the Dissertation

During the process of my dissertation research, I inevitably realized the strong connection between machine learning and high-dimensional data visualization. Each of these two research areas has demonstrated the possibility to have a significant impact on one another. In this dissertation, I have utilized the subspace clustering algorithm, a recently established approach from the machine learning and computer vision community, for capturing important information in high-dimensional space for visualization. On the other hand, I also work with collaborators in natural language processing, utilizing visualization methods to help the domain experts gain an understanding of high-dimensional word embedding space (Word2Vec [129]).

High-dimensional spaces exist in many aspects of the machine learning process. The input of a machine learning algorithm, the feature space, is usually high-dimensional for even the simplest problems. In addition, the learned model usually defines structures or divisions in the high-dimensional feature space (or transformation of such a space). Moreover, outputs of machine learning algorithms can be high-dimensional as well. For example, neural word embedding methods generate high-dimensional spaces that encode semantic relationships. Finally, even the optimization process that often used to build the learning models try to find a local or global minimum of the cost function in a high-dimensional parameter space. One fundamental obstacle that prevents the effective use of machine learning algorithms is the inability to directly understand or detect issues in these high-dimensional spaces, which are opaque for the users. This is particularly prevalent as machine learning proliferates in numerous domains, where non-experts use these processes as black boxes to solve problems in their respective domains. The novice users will likely do not have the expertise for fine tuning parameters or to identify the problems in their learned models. A visualization tool that helps encode the underlying information of machine learning models, which at the same time is easy to understand for novice users, can be extremely usefully in opening the black box and lower the threshold required for effective utilization of machine learning algorithms.

Several visualization approaches have been introduced to aid in the understanding of various machine learning algorithms in the past. Tzeng et al. present a visualization system that helps users design neural networks more efficiently [159]. The works of Teoh and Ma [160] and van den Elzen and van Wijk [161] investigate visualization methods for interactively constructing and analyzing decision trees. Visualization has also been used to aid model validation [162, 163]. However, most of these methods do not directly investigate the high-dimensional aspect of the learning model. Since making sense of the high-dimensional aspect of the machine learning process is essential for understanding why certain model works (or not works) for given data, as a continuation of my dissertation work, I plan to bridge the gap between high-dimensional data visualization and machine learning.

By leveraging high-dimensional visualization approaches, I would like to introduce interactive visual aid for quickly verify or provide a sanity check for the high-dimensional structures in the different computation process. In addition, tracking and understanding the optimization process of the machine learning model is another important aspect for effective utilization of machine learning models. Therefore, I envision the development of a visual debugging tool that provides on-the-fly feedback and monitor of the optimization computation process. In some way, this tool is equivalent of the in-situ visualization often seen in large-scale scientific simulation, where the intermediate computation result or time step is directly visualized in order to understand the simulation process and detect errors early on to save computation resource.

On a grander level, numerous challenges for understanding machine learning algorithms coincide with the goals of high-dimensional data visualization. I believe high-dimensional visualization will play an increasingly important role in designing, tuning, and validating machine learning algorithms.

APPENDIX A

RELATED PUBLICATIONS

- S. Liu, B. Wang, P.-T. Bremer, and V. Pascucci, “Distortion-guided structure-driven interactive exploration of high-dimensional data,” *Computer Graphics Forum*, vol. 33, no. 3, pp. 101–110, 2014

- S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci, “Multivariate volume visualization through dynamic projections,” *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 35–42, Nov 2014

- S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci, “Visual exploration of high-dimensional data through subspace analysis and dynamic projections,” *Computer Graphics Forum*, vol. 34, no. 3, pp. 271–280, 2015

- S. Liu, P. Bremer, J. J. Jayaraman, B. Wang, B. Summa, and V. Pascucci, “The grassmannian atlas: A general framework for exploring linear projections of high-dimensional data,” *Computer Graphics Forum*, vol. 35, no. 3, pp. 1–10, 2016

- D. Maljovec, S. Liu, B. Wang, D. Mandelli, P.-T. Bremer, V. Pascucci, and C. Smith, “Analyzing simulation-based pra data through traditional and topological clustering: A bwr station blackout case study,” *Reliability Engineering & System Safety*, vol. 145, pp. 262–276, 2016

- 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*, vol. PP, no. 99, pp. 1–1, 2016

REFERENCES

[1] L. Wilkinson, A. Anand, and R. Grossman, "Graph-theoretic scagnostics," in *IEEE Symposium on Information Visualization*, vol. 0, 2005, p. 21.

[2] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in *Advances in neural information processing systems*, 2013, pp. 935–943.

[3] R. Clarke, H. W. Ressom, A. Wang, J. Xuan, M. C. Liu, E. A. Gehan, and Y. Wang, "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nature Reviews Cancer*, vol. 8, no. 1, pp. 37–49, 2008.

[4] C. Turkay, P. Filzmoser, and H. Hauser, "Brushing dimensions-a dual visual analysis model for high-dimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2591–2599, 2011.

[5] D. Engel, K. Greff, C. Garth, K. Bein, A. Wexler, B. Hamann, and H. Hagen, "Visual steering and verification of mass spectrometry data factorization in air quality research," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2275–2284, 2012.

[6] D. Maljovec, B. Wang, V. Pascucci, P.-T. Bremer, and D. Mandelli, "Analyzing dynamic probabilistic risk assessment data through topology-based clustering," in *Proceedings of International Topical Meeting on Probabilistic Safety Assessment and Analysis*, 2013.

[7] D. Maljovec, S. Liu, B. Wang, D. Mandelli, P.-T. Bremer, V. Pascucci, and C. Smith, "Analyzing simulation-based pra data through traditional and topological clustering: A bwr station blackout case study," *Reliability Engineering & System Safety*, vol. 145, pp. 262–276, 2016.

[8] S. Gerber, P. Bremer, V. Pascucci, and R. Whitaker, "Visual exploration of high dimensional scalar functions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1271–1280, 2010.

[9] J. Friedman and J. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Transactions on Computers*, vol. C-23, no. 9, pp. 881–890, 1974.

[10] Wikipedia, "Blind men and an elephant — Wikipedia, the free encyclopedia," 2016, [Online; accessed 18-October-2016].

[11] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci, "Visualizing high-dimensional data: Advances in the past decade," *EuroVis State-of-The-Art Report*, 2015.

[12] J. Seo and B. Shneiderman, "A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections," in *IEEE Symposium on Information Visualization.* IEEE, 2004, pp. 65–72.

[13] C.-H. Cheng, A. W. Fu, and Y. Zhang, "Entropy-based subspace clustering for mining numerical data," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 1999, pp. 84–93.

[14] C. Baumgartner, C. Plant, K. Railing, H.-P. Kriegel, and P. Kroger, "Subspace selection for clustering high-dimensional data," in *Fourth IEEE International Conference on Data Mining.* IEEE, 2004, pp. 11–18.

[15] A. Tatu, F. Maas, I. Farber, E. Bertini, T. Schreck, T. Seidl, and D. Keim, "Subspace search and visualization to make sense of alternative clusterings in high-dimensional data," in *IEEE Conference on Visual Analytics Science and Technology.* IEEE, 2012, pp. 63–72.

[16] J. E. Nam and K. Mueller, "Tripadvisor-nd: A tourism-inspired high-dimensional space exploration framework with overview and detail," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 2, pp. 291–305, 2013.

[17] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projections," *IEEE Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, 1997.

[18] L. Wilkinson, A. Anand, and R. Grossman, "High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 6, pp. 1363–1372, 2006.

[19] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. Keim, "Combining automated analysis and visualization techniques for effective exploration of high-dimensional data," in *IEEE Symposium on Visual Analytics Science and Technology.* IEEE, 2009, pp. 59–66.

[20] J. Seo and B. Shneiderman, "Knowledge discovery in high-dimensional data: Case studies and a user survey for the rank-by-feature framework," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 3, pp. 311–322, 2006.

[21] B. J. Ferdosi, H. Buddelmeijer, S. Trager, M. H. Wilkinson, and J. B. Roerdink, "Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators," in *IEEE Symposium on Visual Analytics Science and Technology.* IEEE, 2010, pp. 35–42.

[22] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Processing Magazine*, 2011.

[23] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *2011 International Conference on Computer Vision.* IEEE, 2011, pp. 1615–1622.

[24] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, 2013.

[25] J. Harris, *Algebraic geometry: a first course.* Springer Science & Business Media, 1992, vol. 133.

[26] C. Correa, P. Lindstrom, and P.-T. Bremer, "Topological spines: A structure-preserving visual representation of scalar fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1842–1851, 2011.

[27] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang, "Dis-function: Learning distance functions interactively," in *IEEE Conference on Visual Analytics Science and Technology.* IEEE, 2012, pp. 83–92.

[28] S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci, "Visual exploration of high-dimensional data through subspace analysis and dynamic projections," *Computer Graphics Forum*, vol. 34, no. 3, pp. 271–280, 2015.

[29] S. Liu, B. Wang, J. J. Thiagarajan, P.-T. Bremer, and V. Pascucci, "Multivariate volume visualization through dynamic projections," *IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pp. 35–42, Nov 2014.

[30] S. Liu, P. Bremer, J. J. Jayaraman, B. Wang, B. Summa, and V. Pascucci, "The grassmannian atlas: A general framework for exploring linear projections of high-dimensional data," *Computer Graphics Forum*, vol. 35, no. 3, pp. 1–10, 2016.

[31] S. Liu, B. Wang, P.-T. Bremer, and V. Pascucci, "Distortion-guided structure-driven interactive exploration of high-dimensional data," *Computer Graphics Forum*, vol. 33, no. 3, pp. 101–110, 2014.

[32] K. Ye and L.-H. Lim, "Distance between subspaces of different dimensions," *ArXiv e-prints*, jul 2014.

[33] 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*, vol. PP, no. 99, pp. 1–1, 2016.

[34] T. N. Dang, A. Anand, and L. Wilkinson, "Timeseer: Scagnostics for high-dimensional time series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 470–483, 2013.

[35] D. Guo, "Coordinating computational and visual approaches for interactive feature selection and multivariate clustering," *Information Visualization*, vol. 2, no. 4, pp. 232–246, 2003.

[36] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan, "Selecting good views of high-dimensional data using class consistency," *Computer Graphics Forum*, vol. 28, no. 3, pp. 831–838, 2009.

[37] D. J. Lehmann, G. Albuquerque, M. Eisemann, M. Magnor, and H. Theisel, "Selecting coherent and relevant plots in large scatterplot matrices," *Computer Graphics Forum*, vol. 31, no. 6, pp. 1895–1908, 2012.

[38] A. Inselberg and B. Dimsdale, "Parallel coordinates," in *Human-Machine Interactive Systems.* Springer, 1991, pp. 199–233.

[39] A. Inselberg, *Parallel Coordinates: Visual Multidimensional Geometry and Its Applications.* Springer, 2009.

[40] J. Heinrich and D. Weiskopf, "State of the art of parallel coordinates," *STAR Proceedings of Eurographics*, vol. 2013, pp. 95–116, 2013.

[41] C. Hurley and R. Oldford, "Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions," *Journal of Computational and Graphical Statistics*, vol. 19, no. 4, 2010.

[42] B. J. Ferdosi and J. B. Roerdink, "Visualizing high-dimensional structures by dimension ordering and filtering using subspace analysis," *Computer Graphics Forum*, vol. 30, no. 3, pp. 1121–1130, 2011.

[43] S. Johansson and J. Johansson, "Interactive dimensionality reduction through user-defined combinations of quality metrics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 993–1000, 2009.

[44] M. Novotny and H. Hauser, "Outlier-preserving focus+context visualization in parallel coordinates," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 893–900, 2006.

[45] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen, "Visual clustering in parallel coordinates," *Computer Graphics Forum*, vol. 27, no. 3, pp. 1047–1054, 2008.

[46] R. Rosenbaum, J. Zhi, and B. Hamann, "Progressive parallel coordinates," in *IEEE Pacific Visualization Symposium*, 2012, pp. 25–32.

[47] T. N. Dang, L. Wilkinson, and A. Anand, "Stacking graphic elements to avoid overplotting," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1044–1052, 2010.

[48] W. Peng, M. O. Ward, and E. A. Rundensteiner, "Clutter reduction in multi-dimensional data visualization using dimension reordering," in *IEEE Symposium on Information Visualization.* IEEE, 2004, pp. 89–96.

[49] I. Jolliffe, *Principal component analysis.* Wiley Online Library, 2005.

[50] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky, and R. Chang, "iPCA: An interactive system for pca-based visual analytics," *Computer Graphics Forum*, vol. 28, no. 3, pp. 767–774, 2009.

[51] Y. Koren and L. Carmel, "Visualization of labeled data using linear transformations," in *IEEE Symposium on Information Visualization*, 2003, pp. 121–128.

[52] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[53] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[54] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[55] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[56] M. Williams and T. Munzner, “Steerable, progressive multidimensional scaling,” in *IEEE Symposium on Information Visualization*, 2004, pp. 57–64.

[57] A. Morrison, G. Ross, and M. Chalmers, “A hybrid layout algorithm for sub-quadratic multidimensional scaling,” in *IEEE Symposium on Information Visualization.* IEEE, 2002, pp. 152–158.

[58] A. Morrison and M. Chalmers, “Improving hybrid mds with pivot-based searching,” in *Information Visualization, IEEE Symposium on.* IEEE Computer Society, 2003, pp. 11–11.

[59] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

[60] V. De Silva and J. B. Tenenbaum, “Sparse multidimensional scaling using landmark points,” Technical report, Stanford University, Tech. Rep., 2004.

[61] F. Paulovich, L. Nonato, R. Minghim, and H. Levkowitz, “Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 564–575, 2008.

[62] F. Paulovich, D. Eler, J. Poco, C. Botha, R. Minghim, and L. Nonato, “Piece wise laplacian-based projection for interactive data exploration and organization,” *Computer Graphics Forum*, vol. 30, no. 3, pp. 1091–1100, 2011.

[63] P. Joia, F. Paulovich, D. Coimbra, J. Cuminato, and L. Nonato, “Local affine multidimensional projection,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2563–2571, 2011.

[64] F. Paulovich, C. Silva, and L. Nonato, “Two-phase mapping for projecting massive data sets,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1281–1290, 2010.

[65] M. Gleicher, “Explainers: Expert explorations with crafted projections,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2042–2051, 2013.

[66] J. H. Lee, K. T. McDonnell, A. Zelenyuk, D. Imre, and K. Mueller, “A structure-based distance metric for high-dimensional space exploration with multidimensional scaling,” *IEEE Transations on Visualization and Computer Graphics*, vol. 20, no. 3, pp. 351–364, 2014.

[67] J. A. Lee and M. Verleysen, “Quality assessment of dimensionality reduction: Rank-based criteria,” *Neurocomputing*, vol. 72, no. 7, pp. 1431–1443, 2009.

[68] B. Mokbel, W. Lueks, A. Gisbrecht, and B. Hammer, “Visualizing the quality of dimensionality reduction,” *Neurocomputing*, vol. 112, pp. 109–123, 2013.

[69] T. Schreck, T. von Landesberger, and S. Bremm, “Techniques for precision-based visual analysis of projected data,” *Information Visualization*, vol. 9, no. 3, pp. 181–193, 2010.

[70] C. Seifert, V. Sabol, and W. Kienreich, “Stress maps: analysing local phenomena in dimensionality reduction based visualisations,” in *IEEE International Symposium on Visual Analytics Science and Technology.*, 2010.

[71] C. Turkay, A. Lundervold, A. J. Lundervold, and H. Hauser, "Representative factor generation for the interactive visual analysis of high-dimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2621–2630, 2012.

[72] X. Yuan, D. Ren, Z. Wang, and C. Guo, "Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2625–2633, 2013.

[73] D. Lehmann and H. Theisel, "Optimal sets of projections of high-dimensional data," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 22, no. 1, pp. 609–618, Jan 2016.

[74] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2001, pp. 245–250.

[75] A. Anand, L. Wilkinson, and T. N. Dang, "Visual pattern discovery using random projections," in *IEEE Conference on Visual Analytics Science and Technology.* IEEE, 2012, pp. 43–52.

[76] A. J. Zomorodian, *Topology for Computing (Cambridge Monographs on Applied and Computational Mathematics).* Cambridge University Press, 2005.

[77] S. Biasotti, L. De Floriani, B. Falcidieno, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, and M. Spagnuolo, "Describing shapes by geometrical-topological properties of real functions," *ACM Computing Surveys*, vol. 40, no. 4, pp. 12:1–12:87, 2008.

[78] H. Edelsbrunner and J. Harer, "Persistent homology – a survey," *Contemporary Mathematics*, vol. 453, p. 257, 2008.

[79] H. Edelsbrunner and J. Harer, *Computational Topology - an Introduction.* American Mathematical Society, 2010.

[80] G. Carlsson, "Topology and data," *Bullentin of the American Mathematical Society*, vol. 46, no. 2, pp. 255–308, 2009.

[81] R. Ghrist, "Barcodes: The persistent topology of data," *Bulletin of the American Mathematical Society*, vol. 45, pp. 61–75, 2008.

[82] G. Reeb, "Sur les points singuliers d'une forme de pfaff completement intergrable ou d'une fonction numerique [on the singular points of a complete integral pfaff form or of a numerical function]," *Comptes Rendus Acad. Science Paris*, vol. 222, pp. 847–849, 1946.

[83] S. Smale, "On gradient dynamical systems," *The Annals of Mathematics*, vol. 74, pp. 199–206, 1961.

[84] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci, "Morse-Smale complexes for piece-wise linear 3-manifolds," in *Proceedings 19th Annual symposium on Computational geometry*, 2003, pp. 361–370.

[85] H. Edelsbrunner, J. Harer, and A. J. Zomorodian, "Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds," *Discrete and Computational Geometry*, vol. 30, no. 87-107, 2003.

[86] P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci, "A topological hierarchy for functions on triangulated surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 385-396, 2004.

[87] A. Gyulassy, P.-T. Bremer, V. Pascucci, and B. Hamann, "A practical approach to Morse-Smale complex computation: Scalability and generality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1619–1626, 2008.

[88] A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, and B. Hamann, "Topology-based simplification for feature extraction from 3D scalar fields," in *Proceedings of IEEE Visualization*, 2005, pp. 535–542.

[89] C. Correa and P. Lindstrom, "Towards robust topology of sparsely sampled data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1852–1861, 2011.

[90] M. Nicolau, A. J. Levine, and G. Carlsson, "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival," in *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, 2011, pp. 7265–7270.

[91] G. Singh, F. Memoli, and G. Carlsson, "Topological methods for the analysis of high dimensional data sets and 3d object recognition," in *Symposium on Point Based Graphics*, 2007, pp. 91–100.

[92] G. Sarikonda, J. Pettus, S. Phatak, S. Sachithanantham, J. F. Miller, J. D. Wesley, E. Cadag, J. Chae, L. Ganesan, R. Mallios *et al.*, "Cd8 t-cell reactivity to islet antigens is unique to type 1 while cd4 t-cell reactivity exists in both type 1 and type 2 diabetes," *Journal of autoimmunity*, vol. 50, pp. 77–82, 2014.

[93] S. Schwartz, I. Friedberg, I. V. Ivanov, L. A. Davidson, J. S. Goldsby, D. B. Dahl, D. Herman, M. Wang, S. M. Donovan, and R. S. Chapkin, "A metagenomic study of diet-dependent interaction between gut microbiota and host in infants reveals differences in immune response," *Genome Biol*, vol. 13, no. 4, p. r32, 2012.

[94] J. M. Knight, L. A. Davidson, D. Herman, C. R. Martin, J. S. Goldsby, I. V. Ivanov, S. M. Donovan, and R. S. Chapkin, "Non-invasive analysis of intestinal development in preterm and term infants using rna-sequencing," *Scientific reports*, vol. 4, 2014.

[95] H. Carr, J. Snoeyink, and U. Axen, "Computing contour trees in all dimensions," *Computational Geometry*, vol. 24, no. 2, pp. 75 – 94, 2003, special Issue on the Fourth CGC Workshop on Computational Geometry.

[96] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas, "Robust on-line computation of reeb graphs: Simplicity and speed," *ACM Transactions on Graphics*, vol. 26, no. 3, 2007.

[97] H. Carr and D. Duke, "Joint contour nets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 8, pp. 1100–1113, 2014.

[98] D. Duke, H. Carr, A. Knoll, N. Schunck, H. A. Nam, and A. Staszczak, "Visualizing nuclear scission through a multifield extension of topological analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2033–2040, 2012.

[99] A. B. Lee, K. S. Pedersen, and D. Mumford, "The nonlinear statistics of high-contrast patches in natural images," *International Journal of Computer Vision*, vol. 54, no. 1-3, pp. 83–103, 2003.

[100] B. Wang, B. Summa, V. Pascucci, and M. Vejdemo-Johansson, "Branching and circular features in high dimensional data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 1902–1911, 2011.

[101] V. de Silva, D. Morozov, and M. Vejdemo-Johansson, "Persistent cohomology and circular coordinates," in *Proceedings 25th Annual Symposium on Computational Geometry*, 2009, pp. 227–236.

[102] X. Hu, L. Bradel, D. Maiti, L. House, and C. North, "Semantics of directly manipulating spatializations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2052–2059, 2013.

[103] E. Portes 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 *IEEE Conference on Visual Analytics Science and Technology*. IEEE, 2012, pp. 53–62.

[104] P. Guo, H. Xiao, Z. Wang, and X. Yuan, "Interactive local clustering operations for high dimensional data in parallel coordinates," in *IEEE Pacific Visualization Symposium*, 2010, pp. 97–104.

[105] J. Heer and G. G. Robertson, "Animated transitions in statistical data graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1240–1247, 2007.

[106] D. F. Swayne, D. T. Lang, A. Buja, and D. Cook, "GGobi: evolving from XGobi into an extensible framework for interactive data visualization," *Computational Statistics & Data Analysis*, vol. 43, no. 4, pp. 423–444, 2003.

[107] N. Elmqvist, P. Dragicevic, and J.-D. Fekete, "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1539–1148, 2008.

[108] A. Waddell and R. W. Oldford, "RnavGraph: A visualization tool for navigating through high-dimensional data," 2011.

[109] A. Tatu, F. Maas, I. Farber, E. Bertini, T. Schreck, T. Seidl, and D. Keim, "Subspace search and visualization to make sense of alternative clusterings in high-dimensional data," in *VAST*, 2012, pp. 63–72.

[110] D. Cook, A. Buja, J. Cabrera, and C. Hurley, "Grand tour and projection pursuit," *Journal of Computational and Graphical Statistics*, vol. 4, no. 3, pp. 155–172, 1995.

[111] D. Asimov, "The grand tour: A tool for viewing multidimensional data," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, no. 1, pp. 128–143, 1985.

[112] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," 1973.

[113] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2001.

[114] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2004, pp. 1601–1608.

[115] "Usps handwritten digit dataset," http://statweb.stanford.edu/~tibs/ElemStatLearn/data.html, accessed: 2014-11-30.

[116] D. F. Swayne, D. Temple Lang, A. Buja, and D. Cook, "GGobi: evolving from XGobi into an extensible framework for interactive data visualization," *Computational Statistics & Data Analysis*, vol. 43, pp. 423–444, 2003.

[117] A. Buja, D. Cook, D. Asimov, and C. Hurley, "Computational methods for high-dimensional rotations in data visualization," *Handbook of statistics: Data mining and data visualization*, vol. 24, pp. 391–413, 2005.

[118] R. K. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology,*, vol. 18, no. 3, pp. 259–278, 1969.

[119] D. Kirkpatrick and J. Radke, "A framework for computational morphology," *CG*, vol. 85, pp. 217–248, 1985.

[120] E. R. Hawkes, R. Sankaran, P. P. Pébay, and J. H. Chen, "Direct numerical simulation of ignition front propagation in a constant volume with temperature inhomogeneities: II. Parametric study," *Combust. Flame*, vol. 145, pp. 145–159, 2006.

[121] J. Kniss, G. Kindlmann, and C. Hansen, "Multidimensional transfer functions for interactive volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 270–285, 2002.

[122] H. Guo, H. Xiao, and X. Yuan, "Scalable multivariate volume visualization and analysis based on dimension projection and parallel coordinates." *IEEE transactions on visualization and computer graphics*, 2012.

[123] H. Jänicke, M. Böttinger, and G. Scheuermann, "Brushing of attribute clouds for the visualization of multivariate data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1459–1466, 2008.

[124] D. Whalen and M. L. Norman, "Ionization front instabilities in primordial h ii regions," *The Astrophysical Journal*, vol. 673, pp. 664–675, 2008.

[125] D. Whalen, B. W. O'Shea, J. Smidt, and M. L. Norman, "Photoionization of clustered halos by the first stars," *AIP conference proceedings*, vol. 990, pp. 381–385, 2008.

[126] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[127] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *EMNLP*, vol. 14, 2014, pp. 1532–1543.

[128] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in *HLT-NAACL*, 2013, pp. 746–751.

[129] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[130] M. Faruqui and C. Dyer, "Community evaluation and exchange of word vectors at wordvectors.org," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations.* Baltimore, USA: Association for Computational Linguistics, June 2014.

[131] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[132] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[133] M. Bostock, V. Ogievetsky, and J. Heer, "D^3 data-driven documents," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 12, pp. 2301–2309, 2011.

[134] "Mongodb:cross-platform document-oriented database." https://www.mongodb.org/.

[135] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[136] J. Seo and B. Shneiderman, "A rank-by-feature framework for interactive exploration of multidimensional data," *Information Visualization*, vol. 4, no. 2, pp. 99–113, 2005.

[137] J. Schneidewind, M. Sips, and D. A. Keim, "Pixnostics: Towards measuring the value of visualization," in *IEEE Symposium on Visual Analytics Science and Technology.* IEEE, 2006, pp. 199–206.

[138] F. Mezzadri, "How to generate random matrices from the classical compact groups," *arXiv preprint math-ph/0609050*, 2006.

[139] A. Jennings and J. J. McKeown, *Matrix computation.* Wiley New York, 1992.

[140] D. Cook, A. Buja, and J. Cabrera, "Projection pursuit indexes based on orthonormal function expansions," *Journal of Computational and Graphical Statistics*, vol. 2, no. 3, pp. 225–250, 1993.

[141] E.-K. Lee, D. Cook, S. Klinke, and T. Lumley, "Projection pursuit for exploratory supervised classification," *Journal of Computational and Graphical Statistics*, vol. 14, no. 4, 2005.

[142] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen, "Data visualization with multidimensional scaling," *J. Comp. Graph. Stat.*, vol. 17, no. 2, pp. 444–472, 2008.

[143] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker, "Visual exploration of high dimensional scalar functions," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1271–1280, 2010.

[144] P. Horton and K. Nakai, "A probabilistic classification system for predicting the cellular localization sites of proteins." in *Ismb*, vol. 4, 1996, pp. 109–115.

[145] L. Saul and S. Roweis, "Think globally, fit locally: unsupervised learning of nonlinear manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, 2003.

[146] A. Gorban and A. Zinovyev, "Principal manifolds and graphs in practice: from molecular biology to dynamical systems," *Int. J. Neural. Syst.*, vol. 20, no. 3, pp. 219–232, 2010.

[147] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[148] J. Venna, J. Peltonen, K. Nybo, H. Aidos, and S. Kaski, "Information retrieval perspective to nonlinear dimensionality reduction for data visualization," *J. Mach. Learn. Res.*, vol. 11, pp. 451–490, 2010.

[149] S. France and D. Carroll, "Development of an agreement metric based upon the rand index for the evaluation of dimensionality reduction techniques, with applications to mapping customer data," *Machine Learning and Data Mining in Pattern Recognition, Lect. Notes Comput. Sc.*, vol. 4571, pp. 499–517, 2007.

[150] M. Aupetit, "Visualizing distortions and recovering topology in continuous projection techniques," *Neurocomputing*, vol. 70, pp. 1304–1330, 2007.

[151] B. Gärtner, "Fast and robust smallest enclosing balls," *Algorithms-ESA'99, Lect. Notes Comput. Sc.*, pp. 325–338, 1999.

[152] A. Agarwal, J. M. Phillips, and S. Venkatasubramanian, "Universal multi-dimensional scaling," *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 1149–1158, 2010.

[153] L. Cayton and S. Dasgupta, "Robust Euclidean embedding," *Proc. Int. Conf. on Machine learning*, pp. 169–176, 2006.

[154] D. Defays, "An efficient algorithm for a complete link method," *Comput. J.*, vol. 20, no. 4, pp. 364–366, 1977.

[155] S. Gerber, P.-T. Bremer, V. Pascucci, and R. T. Whitaker, "Visual exploration of high dimensional scalar functions," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1271–1280, 2010.

[156] S. Lisitsyn, C. Widmer, and F. J. I. Garcia, "Tapkee: An efficient dimension reduction library," *J. Mach. Learn. Res.*, vol. 14, pp. 2355–2359, 2013.

[157] D. Mandelli, A. Yilmaz, T. Aldemir, K. Metzroth, and R. Denning, "Scenario clustering and dynamic probabilistic risk assessment," *Reliab. Eng. Syst. Safe.*, vol. 115, pp. 146–160, 2013.

[158] J. W. Tukey, "Exploratory data analysis," 1977.

[159] F.-Y. Tzeng and K.-L. Ma, "Opening the black box - data driven visualization of neural networks," in *Proceedings of IEEE Visualization*, 2005, pp. 383–390.

[160] S. T. Teoh and K.-L. Ma, "Paintingclass: interactive construction, visualization and exploration of decision trees," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2003, pp. 667–672.

[161] S. van den Elzen and J. van Wijk, "Baobabview: Interactive construction and analysis of decision trees," in *IEEE Conference on Visual Analytics Science and Technology*, 2011, pp. 151–160.

[162] P. Rheingans and M. desJardins, "Visualizing high-dimensional predictive model quality," in *Proceedings of IEEE Visualization*, 2000, pp. 493–496.

[163] M. Migut and M. Worring, "Visual exploration of classification models for risk assessment," in *IEEE Symposium on Visual Analytics Science and Technology*, 2010, pp. 11–18.