UNIVERSITY OF CALIFORNIA
SANTA CRUZ

INTERACTIVE HIGH-DIMENSIONAL DATA ANALYSIS USING THE "THREE EXPERTS"

A thesis submitted in partial satisfaction of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Georg H Albrecht

March 2015

The Thesis of Georg H Albrecht
is approved:

Professor Alex Pang, Chair

Professor Suresh Kumar Lodha

Professor Sri Kurniawan

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Interactive High-Dimensional Data Analysis using the “Three Experts”

by

Georg H Albrecht

With the increasing availability of different kinds of data from various domains such as health care, finance, social networks, etc. there is a need to provide analytic tools that are more accessible to lay people. In this paper, we present a software tool which can be used to help make high dimensional data understandable for inexperienced users. To facilitate the understanding of the data, we place special emphasis on how the data is presented, using the “Three Experts”, and on showing personalized information within the data. The Three Experts display shows the results of three different dimension reduction techniques, similar in notion to seeking several expert opinions regarding a particular topic. This will help the user to discern between pertinent structures in the data, and those resulting from the distortion inherent in dimension reduction. The second emphasis is on providing the ability for users to identify, insert, and manipulate points of interest among the sea of data. In addition, the user can observe high dimensional trajectories from one position in the data set to another. This will help convey the changes necessary to displace a point to its desired target state. Observing these changes will enable the user to develop an actionable intuition for the data in question. Though the currently envisioned application for such a system is in health care, such methods could potentially be used in any field where high dimensional data needs to be analyzed.

Acknowledgments

I would like to thank my father Georg F Albrecht, and grandfather Georg L Albrecht, for their unwavering support of my academic endeavors. I would also like to thank my adviser Prof. Alex Pang for his guidance and patience.

Chapter 1

Introduction

The continued proliferation of complex data sets across various domains has created a demand for both automatic and human-in-the-loop methods that can be used to extract actionable information. This process, called data mining or knowledge discovery, has been the subject of much research. As the amount of accessible data continues to increase, so will the demand for new and improved methods with which to process this data. For most high-dimensional data sets, domain experts can make use of various software tools which aid the observation, inspection, and analysis of the data in question. Visual data mining tools with interactive presentation and query capabilities allow domain experts to quickly examine complex data while interacting with multivariate visual displays. In addition, researchers are realizing that visual feedback has a role to play in the data mining process, as well as the analysis of the results. The ability to create a good mental model of how high dimensional data is structured is essential if end users expect to develop a sound understanding of the data. Unfortunately, many of these software tools are intended for use by professionals who are likely already familiar with high dimensional data. However, there are many situations in which users, who are otherwise unfamiliar with high dimensional data, could benefit from exposure to, and exploration of, such data.

In this paper we present a software tool designed to help make high dimensional data understandable to users who are otherwise inexperienced or unfamiliar with such information. This is done by providing a simplified visual analytics platform to enable exploration of and interaction with a particular data set in a controlled fashion. There are two main aspects to this software. The first is the use of the ‘three experts’ display. The ’three experts’ display provides the user with three views of the data. These three views corresponds to three different dimension reduction techniques, which will likely produce similar, but not identical results. Because a user may not be familiar with high dimensional data, much less dimension reduction techniques, these views are meant to act as ‘expert interpretations’ of the data. By comparing these three views, the user can develop a good sense for which structures (such as groupings and spacings) are consistent between views, and which are not. The second important aspect of the proposed software is the exploratory features made available to the user. These exploratory features allow the user to gain an intuition for the data by directly observing the effects of their manipulations. To this end, the user is able to create a new data point, based on attribute values unique to the user. Once created, the user can observe the new point’s position within the data set. In addition, the user can view the movement of the unique point resulting from manipulation of the point’s attribute values. Finally, the user is also able to create various trajectories which help to show the changes necessary to re-position a point.

The main motivation for this research is in the field of health care. Such a system could enable doctors and patients to determine potential courses of prevention or treatment of a condition, based on the specific attributes of the patient. Moreover it would allow a patient to view their position, based on their unique attributes, relative to others with a similar condition, spanning a range from afflicted to healthy. Doing so would help the patient to see what changes may be necessary for them to reach a more desirable state of health.

An overview of the intended use for this software can be best illustrated by example. Given a data set with (ideally) two distinct clusters, healthy and sick, a user is able to create a new data point, based on their unique health attributes. This new data point is then inserted into the original data set allowing the patient to see their position within the larger data set. The software shows various indications of how changes to some attributes will cause little or no alteration to the patient's position, while changes to other attributes, or combinations thereof, will result in larger spatial displacements. Using this information the user can then manipulate the various attributes, which represent aspects of their health and lifestyle, and see how these changes affect the position of their data point relative to the original data set. The result is a more informed patient with a better understanding of their current condition, and the potential results of altering various aspects of their current lifestyle. While this example stresses a potential medical application, such a system could be used in other fields, such as the sciences, finance, and business, where high dimensional data needs to be analyzed, but the user may not be familiar with high dimensional data.

The remainder of the paper is organized as follows: Prior work done in several related areas is discussed in section 2, as well as the sources for various techniques used. Section 3 discusses the main interface design considerations as well as an overview of the software itself. Section 4 offers examples of potential usage with a well known data set. Section 5 reviews the results of preliminary user feedback, and section 6 addresses some shortcomings as well as areas for improvement. Finally, section 7 provides concluding remarks, as well as potential areas for improvement.

Chapter 2

Previous Work

Humans are not directly capable of visualizing information or structured objects in more than three dimensions without some form of abstraction or manipulation. In an effort to overcome this hindrance much research has been done to make higher dimensional data more perceivable and understandable. Below we provide an overview of the main areas of research on this subject, both in general, and with direct medical applications.

2.1 N-D Projection

Early attempts at visualizing higher dimensions were often concerned with structured objects, instead of data, such as the higher dimensional analog to the cube, the hyper-cube. This research often focused on the fourth dimension, due to the relative ease of visual understanding, and because these methods could be extended to higher dimensions. Examples of such investigations include “A Computer Technique for Displaying n-Dimensional Hyperobjects” by Michael Noll [18], and the thesis by Steven Richard Hollasch, “Four-Space Visualization of

4D Objects” [21]. Research in this area is still taking place, as with the work done by Sakai and Hashimoto [19] which explores methods for mapping 4-D structures resulting from complex mathematical functions, and movement within this 4-D space from the users’ perspective. However it should be noted that these sources deal mainly with the projection of high dimensional structures. They do not focus on methods with which to explore and interact with high dimensional data sets.

2.2 Dimension Reduction

When dealing with high dimensional data (as opposed to structural objects or shapes), the high number of attributes can make working with, or visualizing, the data an unwieldy and resource intensive task. It should be noted that the term “attribute” is representative of a corresponding dimension within the data. To make the data more amicable, one is often able to transform the high dimensional data into an adequately representative form with fewer dimensions. Sometimes this can be done by choosing to include or exclude particular features, thereby creating a chosen subset of features, and therefore dimensions. This process is often referred to as feature selection.

Another form of dimension reduction is called feature extraction. As the number of dimensions increases, so does the likelihood for correlations between the various attributes, and therefore the dimensions. Feature extraction works by seeking out these redundant attributes and representing the data using a new set of non-redundant attributes. While many dimension reduction methods exist [8], we make use of Principal Component Analysis, Singular Value

Decomposition, and Multi-Dimensional Scaling.

Principal component analysis (PCA) [4] is used to transform potentially correlated attributes into a smaller number of uncorrelated attributes, called the principal components. These principal components serve as the new axis of the lower dimensional representation, and are ranked according to the variability present along each newly defined axis. Additionally, while information loss is inherent, the user can control the trade off between this loss and the reduction in dimensions.

Another potential method of dimension reduction is a low-rank approximation of Singular Value Decomposition (SVD) [12]. Given a data matrix M, singular value decomposition of a M results in three matrices commonly represented as, U, $\sum$, and V, where $M = U * \sum * V$. The columns within matrices U and V can be treated as orthogonal unit vectors, while as a whole both matrices represent rotations. The matrix $\sum$ is a diagonal matrix. The values along the diagonal of $\sum$ are called singular values and are ranked according to their contributions. To perform a low-rank approximation involved removing the singular values with smaller contributions while keeping those with higher contributions.

If one is primarily concerned with visualizing the data, and less concerned with the formation of an underlying representative model, then Multi-Dimensional Scaling (MDS) [14] is a possible solution. The goal of MDS is to provide a low dimensional visual representation of data which maintains the distances observed among the points in higher dimensions. To achieve this, MDS uses a function minimization algorithm which evaluates different configurations of points in an attempt to minimize the disparity in distance between the original and lower dimensional representations.

2.3 Interactive High-D Data

Once the various aspects of high dimensional data projection and visualization became established, the next step was to facilitate the exploration of, and interaction with, the data. This was often done alongside some form of data mining so that researchers could inspect the resultant clusters, outliers, or association rules, more closely. There are many proprietary and open source software packages available for visual data analytics. Examples of these include Tableau [20], Microsoft Business Analytics [16], ggobi [9], and XmdvTool [3]. A notable software package is "ClusterSculptor" by Nam et.al. [17], which allows for interactive tuning of clustering parameters, principal components, and other aspects of the data. However, these systems do not allow for the insertion and manipulation of user specific data points, nor do they provide an interactive way to show the changes necessary to displace various points within the data.

2.4 Projection Pursuit

When projecting from a higher to lower number of dimensions, information may be displayed from a viewpoint that obscures interesting features within the data. This led to the development of a technique called Projection Pursuit, which was first successfully implemented by Friedman and Turkey [10]. The objective of projection pursuit is to find informative projections of high dimensional data. The measure of the "interesting" projections being pursued is based on some optimizable criteria, such as a large deviation from other projections, implying a feature of interest. An interactive variation on this technique is Targeted Projection Pursuit, which was developed by Faith, Mintram, and Angelova [6] [5]. This is done by allowing the user to first specify their intended target projection. Within a 2D projection, the user moves the points shown to a certain position of their liking, creating the target. The algorithm then performs a search for a projection that closely matches the user specified target projection, and returns the highest ranking result. While these methods can be used to uncover interesting or user-specified features within high dimensional data, they are not able to form a lower dimensional representation of the data in question.

2.5 Current Medical Applications

Work with direct medical application has been performed as well, such as the lung cancer outcome calculator by Agrawal et.al.[1]. Here, Agrawal analyzed a lung cancer data set, using an ensemble of decision tree based classifiers, in an effort to develop accurate survival prediction models for respiratory cancer. This method was able to accurately estimate a lung cancer patient's chances of survival, in a time frame of nine months to two years. The authors suggest that such software can also be used to provide the patient with a better understanding of the risks involved in various treatments, based on patient-specific attributes.

A recent review paper by Garcia-Retamero et.al. [11] highlights research showing that many patients experience difficulties understanding numerical concepts which are relevant to their health. The paper discusses several examples of how patients often weigh certain factors more highly than others (referred to as "denominator neglect") resulting in less consideration for other important information. Issues that are discussed include factors such as the lack of numerical or language-related skills, and the extent to which these deficiencies can be mitigated with the use of visual aids.

Chapter 3

User Interface

The main goal of this project was to create an interface which will enable an otherwise inexperienced user to gain an intuitive understanding of high dimensional data. To achieve this requires the proper balance of exposing the user to the complexity inherent within the data, and taking care not to overwhelm them with intricate details. The user will likely not benefit from exposure to the procedures used to process and present the data. However, if the user is to gain an actionable intuition for the data, controlled exposure to the defining aspects of this complexity is imperative. Finding the appropriate balance between shielding and exposure is what makes this task challenging.

With this in mind, we formulated the following functional requirements as necessary to properly facilitate the exploration and manipulation of high dimensional data. These requirements are *a)* provide three visual "expert interpretations" of the data, each using different dimension reduction techniques; *b)* show the relative distributions of the data on a per attribute basis; *c)* allow the creation and modification of new user specified data points; and *d)* allow the creation of trajectories to demonstrate the changes necessary to displace a point from its current position, to a new user specified target position. These four functional requirements work together to enable controlled exploration of, and interaction with, the data. Each helps provided information about the salient features of both the individual attributes, and the data set as a whole.

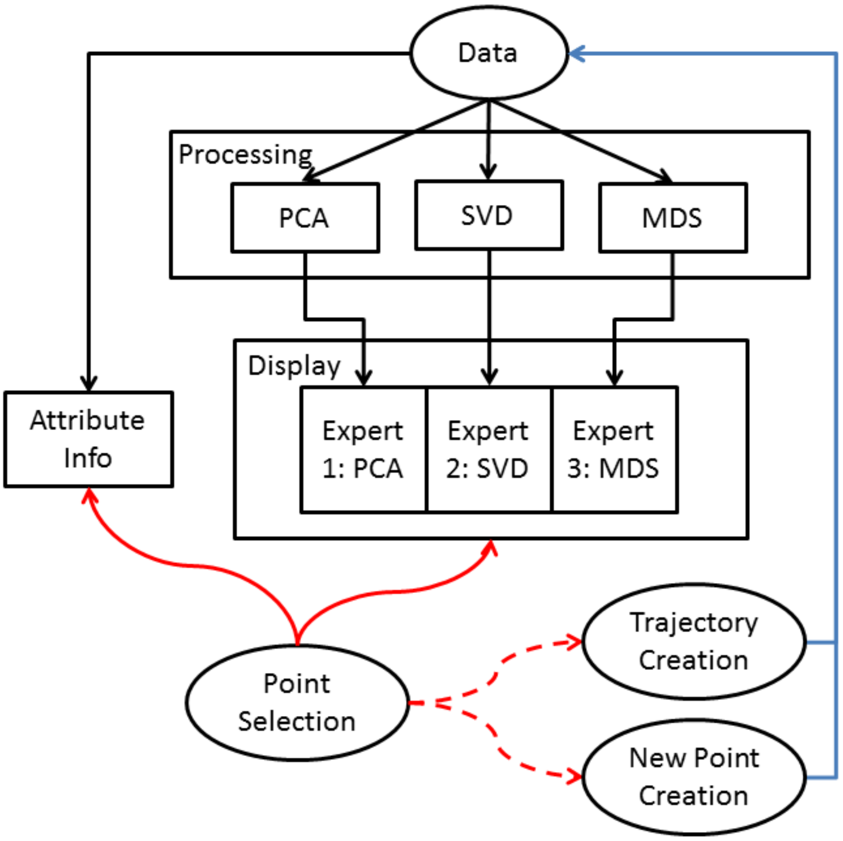


Figure 3.1: The basic software architecture. Black lines indicate the data flow within the program, red lines indicate user selections (dashed lines are optional), and blue lines indicate modifications to the data set. The data is processed using three dimension reduction techniques, which are displayed as three “Expert Interpretations”. The user has three forms of interaction: Point selection, which is linked to the Attribute Info histograms; Point Creation, which inserts a new point into the data set; and Trajectory creation, which shows the changes necessary to displace a point to a desired target position.

In general, the interface is broken into four distinct groupings of components. These groupings are consistent with the four functional requirements stated above. An example of the default interface can be seen in Fig. 3.2. In this figure, the three data displays showing the "expert interpretations" are seen in the top right. These displays are always visible to the user, regardless of the actions being taken. The bottom consists of a tabbed pane with each tab labeled according to a particular task or form of information it shows. The default visible panel shows the basic information for each attribute present in the data. The interface itself has been constructed with components which should be familiar to anyone who has used a computer. There are no custom interface components, thus helping to promote a baseline affordance for new users, who will not need to spend extra time learning new interface components. The following sections will provide more discussion of the various functional requirements, as well as their implementation details.

3.1 The Three Experts

Once a data set of interest has been loaded, the user is shown three "expert interpretations" of the data. These interpretations correspond to three different dimension reduction techniques: Principal Component Analysis, Singular Value Decomposition, and Multi-Dimensional Scaling. Since the user will likely not be familiar with (or interested in) the methods used, the representations are instead referred to as "expert interpretations". This choice of nomenclature is intended to suggest the notion of seeking a second or third opinion by consulting alternative experts in a particular field. The three specified techniques were chosen because of their perceived popularity when low dimension (in our case 2D) visualization of higher dimensional data is required. Regardless of their popularity, these techniques, like any dimension reduction techniques, have an implicit side effect of distortion of the data resulting from information loss.

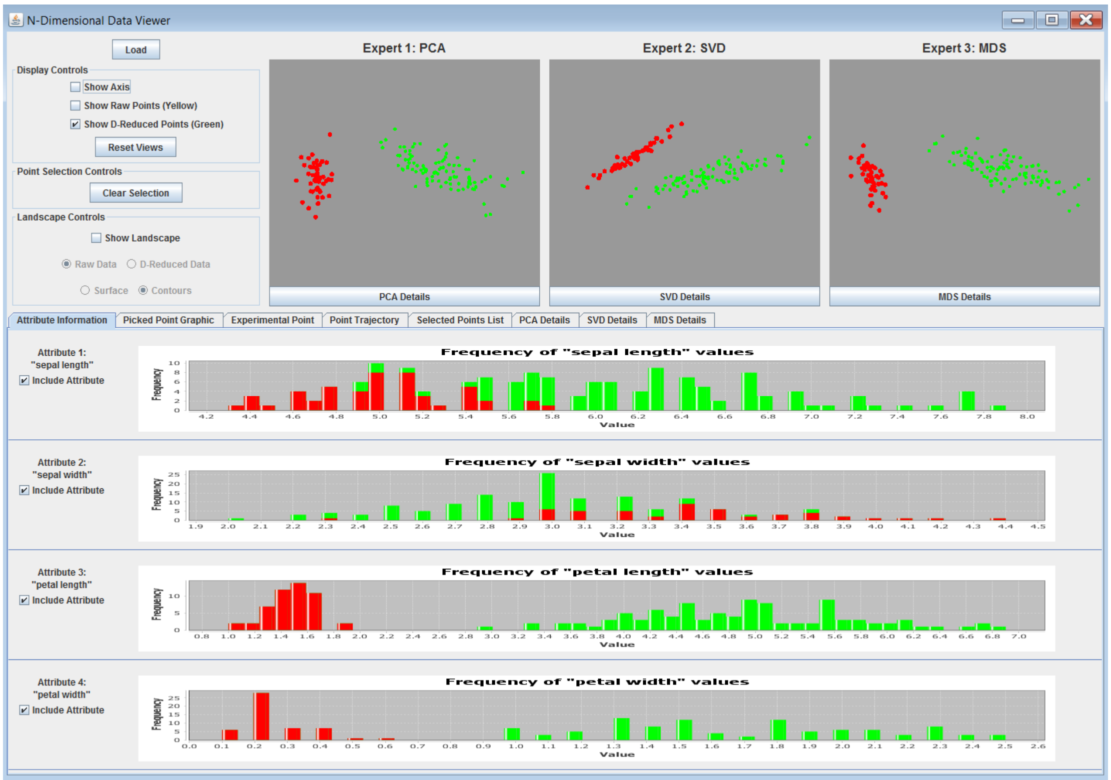


Figure 3.2: The default user interface. The Iris data is being shown according to three different dimension reduction techniques. The bottom tab shows a histogram indicating the distribution of attribute values. The "include attribute" check boxes allow the user to dynamically remove attributes from the data set. A selection of points in the data set is shown in red. The histograms below are linked to the selection and show the distribution of the attribute values for the selection.

Therefore the purpose of presenting the results of three different dimension reduction methods is to help mitigate potential misconceptions about the data as a result of the distortion (e.g. bias of an expert). Observation of the similarities and differences between the three views will show the user how open to interpretation various structures are. Each form of dimension reduction works to represent the data in a 2D space while retaining its important characteristics, but each uses an alternate approach by considering different aspects of the data. For example, PCA seeks to preserve variability, SVD attempts to capture mathematical relationship between the rows and columns of the data matrix, while MDS works to maintain the distances between each point. More details about these methods is provided below.

With Principal Component Analysis (PCA), one is able to represent the same data set using fewer dimensions, while limiting the loss of information. This is done by transforming potentially correlated dimensions into fewer non-correlated dimensions, while maintaining the variability within the data. However the resulting orthogonal dimensions no longer represent a single attribute. Instead, each resulting axis is now representative of varying weighted contributions from the original attributes. One advantage of PCA is that a new point can be transformed without the need to recalculate the principal components. This is not the case for the other two chosen methods.

Singular Value Decomposition (SVD) is a form of matrix decomposition, the result of which is three matrices. Two of the resulting matrices represent rotations, while the third can be regarded as a scaling matrix. This scaling matrix contains the singular values along the diagonal of the matrix. These singular values are unique to the original data, and can be ranked according to the severity of the scaling. Dimension reduction with SVD can be achieved by performing a low-rank approximation of the original data. This low-rank approximation works by keeping only the desired number of highest ranked singular values (in our case only the top 2) and replacing the lower ranked singular values with zero. The resulting rank reduced data set now only contains the desired number of dimensions.

The third method, Multi-Dimensional Scaling (MDS), is fundamentally different from the previous two dimension reduction methods. MDS works by taking the data set in question and producing a representation of the data in a lower dimension. The resulting lower dimension representation is optimized to maintain the relative distances observed between the points in higher dimensions. That is to say, MDS aims to place each point in a lower dimensional space such that the relative distances between each point is preserved as much as possible. Because the algorithm only considers the distance between points, there are several important factors to be aware of. First, as the axes of the resulting lower dimensional representation are not scaled, they become arbitrary, and may change as the data changes. Second, the orientation of the picture is arbitrary, and can change as a result of changes in the data set. Third, while the formation and spacing of larger distances between clusters will be well represented, the tighter spacing of points within clusters will be less accurately represented.

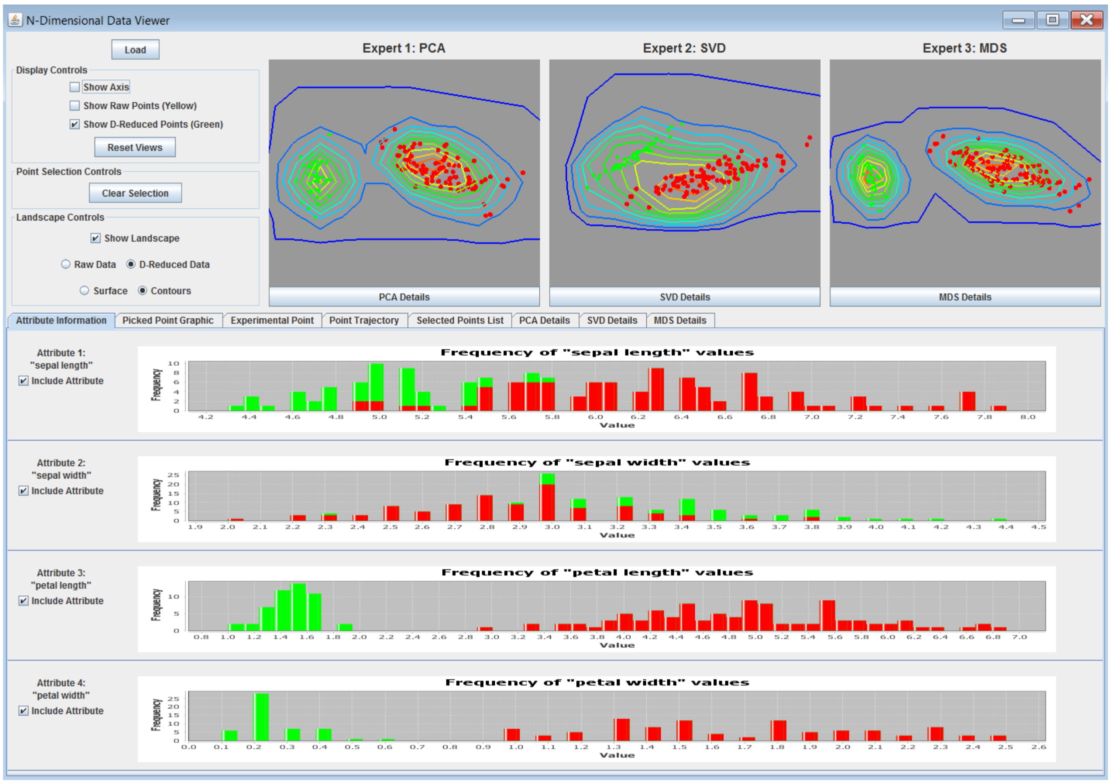
The resulting representations should all be able to capture the most salient structures, but may differ in the representation of smaller features. This allows the user to visually filter the structures within the data for the user. If similar clusters, shapes, or spacing, are present in the majority of views, then it serves as a good indication these structures are inherent in the data and may be worth investigating further. However, if there is a seemingly indicative cluster, shape, or spacing present in only one of the views, then it is more likely an artifact of a particular dimension reduction technique, and not a feature present within the data.

3.2 Data Inspection and Creation

Two of the key functional requirements for this user interface were the ability to view the distribution of the data, as well as user specified subsets, and the ability to create and insert a modifiable point into the data set.

The inspection ability is provided by the default Attribute Information tab, shown in Fig. 3.2. This shows the user the range over which the attributes values span, as well as the distribution of values across this range in the form of a histogram. The user also has the ability to select a subset of points within the expert displays for closer inspection. Points can be selected individually, or by dragging a selection box over points of interest. The selected points are then highlighted across all three expert displays, and are linked to the attribute histograms. Thus, the user can quickly see the variation and distribution of the subset of interest along each attribute, both within the selection, and relative to the entire data set. In addition, the user is allowed to disable individual attributes as they see fit (though at least two must be enabled for display purposes). The user can then perform all the same tasks as if the data only consisted of the selected attributes.

A potential artifact when forming lower dimensions is that points, which are spatially separated at higher dimensions, may project to the same location in lower dimensional space. To help the user recognize instances of high point density, we included the ability to display a density map, using either contours or a landscape, for each expert. An example of the contour maps can be seen in Fig. 3.3. The density map will convey to the user the obvious point concentrations. More importantly with inspection the user can identify areas that may appear to contain only a few points when in fact there are many, or areas that appear to contain many points, when in fact there are only very few. Upon selection of one of these seemingly single points, all the points sharing that location in the low dimension space will be distinguishable by their variations in higher dimensions, shown in the per attribute display.


N-Dimensional Data Viewer

Display Controls
□ Show Axis
□ Show Raw Points (Yellow)
☑ Show D-Reduced Points (Green)

Landscape Controls
☑ Show Landscape
○ Raw Data ● D-Reduced Data
○ Surface ● Contours

Attribute 1: "sepal length" ☑ Include Attribute
Attribute 2: "sepal width" ☑ Include Attribute
Attribute 3: "petal length" ☑ Include Attribute
Attribute 4: "petal width" ☑ Include Attribute

Figure 3.3: A density based contour map of the Iris data set. This is being used to check for point densities which are higher or lower than the density of the visible points. This may result when mapping data from from high to low dimensions. Upon inspection we can see that the highest point densities occur in the middle of the larger cluster, in all three views.

Allowing the user to create and manipulate their own unique data point is paramount. The user can create a point using attribute values derived from a preexisting point, as a mean representation of a particular cluster of interest, based off of personal traits, or simply for experimental ”what-if” scenarios. This point creation and manipulation is enabled by the Experimental Panel. Values for each attribute can be specified one of two ways: The user can either enter the numeric values themselves, or they can select a preexisting point or group of points as the basis for the new point. Numerically specified attribute values will more likely be used if a point is being created to represent the user, such as in the previously described medical situation. On the other hand, selection of a predefined point or cluster will more likely be used to investigate some aspect of the data itself. If using a selection from the data set, the attribute values will be derived from the average attribute values of the selected points. Regardless, once all necessary attribute values have been specified, the new point can then be inserted. This insertion requires applying the three dimension reduction techniques to the newly-expanded data set and displaying the respective results.

3.3 Data Manipulation

Furthermore, once the user specified point has been created, the user is able to manipulate the various attribute values as they see fit. Manipulating these values will allow the user to directly observe the degree to which changes to particular attributes will affect the position of the experimental point, within the original data set. This manipulation is done either by moving the value slider or by entering a new value for the appropriate attribute. As changes are made, the position of the experimental point is updated in real time across all three expert displays. Though computationally expensive (as SVD and MDS do not allow for dynamic addition of points and so must be recomputed), this real-time display enables direct feedback for the user.

In the context of the previously discussed medical application, the user would enter their pertinent medical information to create a new point with attribute values based on their own unique health characteristics. Once displayed, the user would then see where their point lies in space relative to the data set of healthy and unhealthy people. From here, the user could modify various personal health related attributes and observe how these changes alter the position of their unique point. Consequently, the user will see and better understand what alterations to their current state would be necessary to reach a position indicative of improved health.

3.4 Point Trajectory

However, the changes necessary for a user created point to reach a desired position may not be straightforward to the user. To help overcome this obstacle, the user can activate a form of guided interpolation along a trajectory. This guided interpolation will show the user how the values of each attribute will need to change in order to displace a point from its current position to a user specified target position. Both the start and end points can be specified in the same manner as the initial experimental point. The user can either specify them numerically, or as separately selected points or grouping of points. Some attributes, such as age or sex, may not be strictly modifiable and so should not be considered when creating a trajectory. The user may want to disable these attributes, or any others they deem modifiable, in the attribute information panel. This will exclude the disabled attributes from the interpolation trajectory.

The user has a choice of the type of interpolation used to create the trajectory: linear or greedy. As implemented, linear interpolation works by independently performing a discrete number of linear interpolation steps along each attribute. The resulting points along the trajectory represent a simultaneous progression across all attributes. This progression results in a straight line between the start and end points in high dimensional space. However it may be difficult for the user to keep track of the changes along each attribute simultaneously. To help overcome this the user may instead make use of a greedy interpolation trajectory. Greedy interpolation performs a discrete number of interpolation steps along each attribute, individually. The greedy nature of this form of interpolation is due to the algorithm always choosing to interpolate along the attribute with the smallest absolute change in value between its start and end points. Because the greedy interpolation focuses on progressing one attribute at a time, the final trajectory will be composed of as many line segments as there are attributes. The initial segments will be short, representing the smaller change in attribute value, while the later segments will be longer, representing the larger changes necessary.

After activating the trajectory of their choice, the user is able to step through the intermediate points along the trajectory. The current point is identifiable as the largest step marker along the displayed trajectory, and will change as the user progresses along it. As the user continues stepping along the trajectory, the progression of the currently changing attributes is shown in their respective progress bars. This provides the necessary feedback to allow the user to easily follow the changes along the various attributes as they occur. For a linear interpolated trajectory the progression will occur evenly across all attributes, whereas for greedy interpolation the progression will occur one attribute at a time.

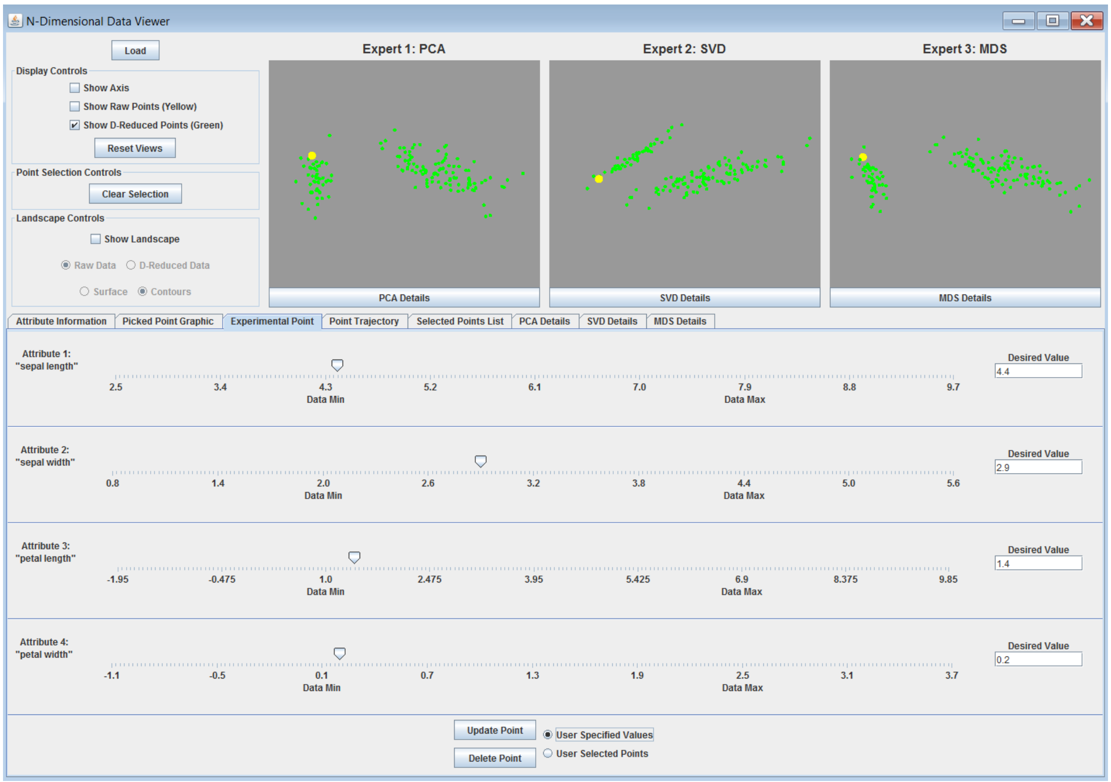
Chapter 4

Data Set Exploration

In this section we will demonstrate the usage of our software with the canonical Iris data set [7]. The Iris data set was chosen because it is well-studied and contains two distinct clusters. Using this dataset allows us to verify that the methods we apply are performed correctly, and that the program's results are predictable. The Iris data set example will showcase the inspection of particular attribute characteristics, and how they relate to the larger data set.

4.1 Iris Data Set

The Iris data set contains 150 samples, with four attributes per entry. The attributes are measurements of the length and the width of the sepals and petals of three different Iris species. Fig. 3.2 shows the software with the Iris data set displayed. The default tab shown is the attribute information tab. The histograms displayed under this tab are linked to the point selections made by the user. Once a point or group of points is selected, the distribution of the selected points is shown on a per attribute basis. In the prior figure, the user has selected a cluster of points, which are shown in red in both the expert displays and the linked histograms.



□ Show Axis

□ Show Raw Points (Yellow)

☑ Show D-Reduced Points (Green)

□ Show Landscape

● Raw Data ○ D-Reduced Data

○ Surface ● Contours

Attribute 1: "sepal length" Desired Value 4.4

Attribute 2: "sepal width" Desired Value 2.9

Attribute 3: "petal length" Desired Value 1.4

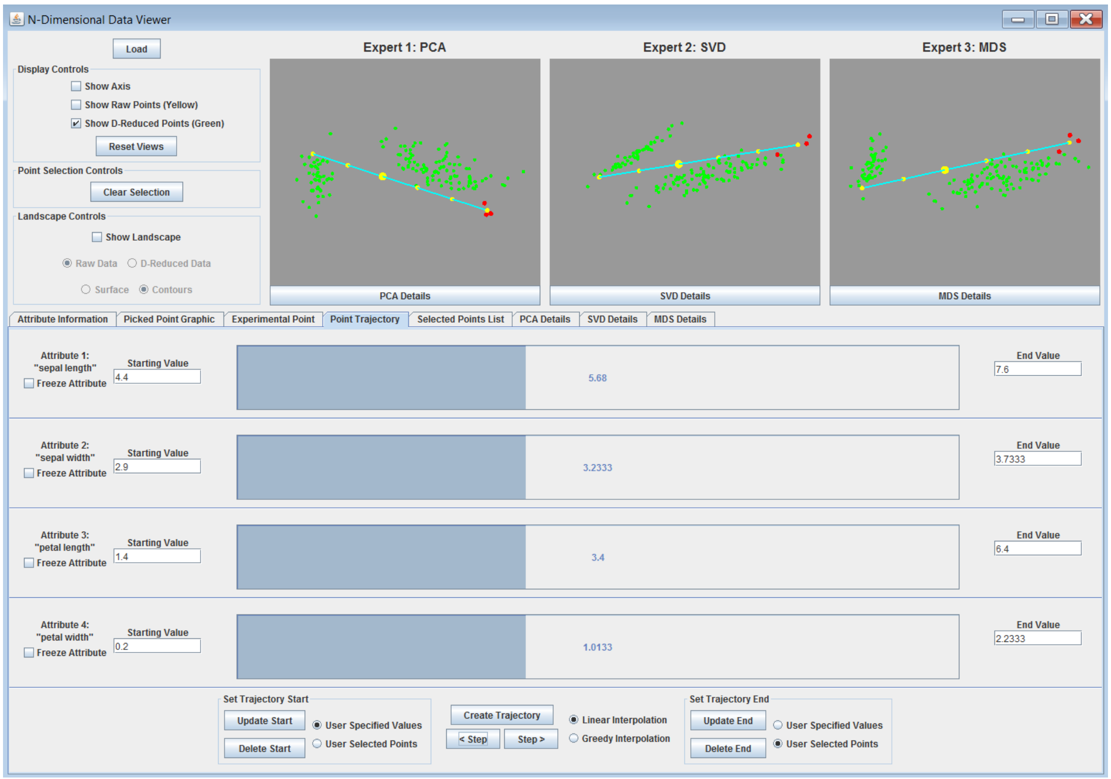
Attribute 4: "petal width" Desired Value 0.2

● User Specified Values

○ User Selected Points

Figure 4.1: The Experimental Point user interface tab. The Iris data is shown in green according to three different dimension reduction techniques. The newly created point is shown in yellow. The bottom tab shows the sliders used to control the value of the experimental point. The three displays are updated in real time according to changes in value.

For the sake of an example, assume a user decides to insert a new experimental point, perhaps based on values from a newly-collected specimen. To create this point, the user selects the Experimental Point tab. Here the attribute values can be entered by typing the values directly, by adjusting the value sliders, or by using the mean attribute values of a user-selected point or points. Once the values for the respective attributes have been entered, the new point is inserted into the original data set and the three dimension reduction methods are applied. An example of the newly created point can be seen in Figure 4.1 as the larger yellow point amid the smaller green points within the respective expert views. From here the user can make changes to the various attribute values. Altering these values with the sliders will allow the user to observe the resulting displacement in real time. The real-time displacement serves to convey to the user how each attribute contributes to the position of the data point. With this ability the user could attempt to move the experimental point to a particular target area; however this can be difficult. Because there are many attributes which affect the position of a point, there may be many attribute value combinations resulting in a visual proximity of the experimental point to the target area.



N-Dimensional Data Viewer

Display Controls
☐ Show Axis
☐ Show Raw Points (Yellow)
☑ Show D-Reduced Points (Green)

Landscape Controls
☐ Show Landscape
● Raw Data ○ D-Reduced Data
○ Surface ● Contours

Attribute 1: "sepal length" ☐ Freeze Attribute — Starting Value 4.4 — 5.68 — End Value 7.6

Attribute 2: "sepal width" ☐ Freeze Attribute — Starting Value 2.9 — 3.2333 — End Value 3.7333

Attribute 3: "petal length" ☐ Freeze Attribute — Starting Value 1.4 — 3.4 — End Value 6.4

Attribute 4: "petal width" ☐ Freeze Attribute — Starting Value 0.2 — 1.0133 — End Value 2.2333

Set Trajectory Start: ● User Specified Values ○ User Selected Points

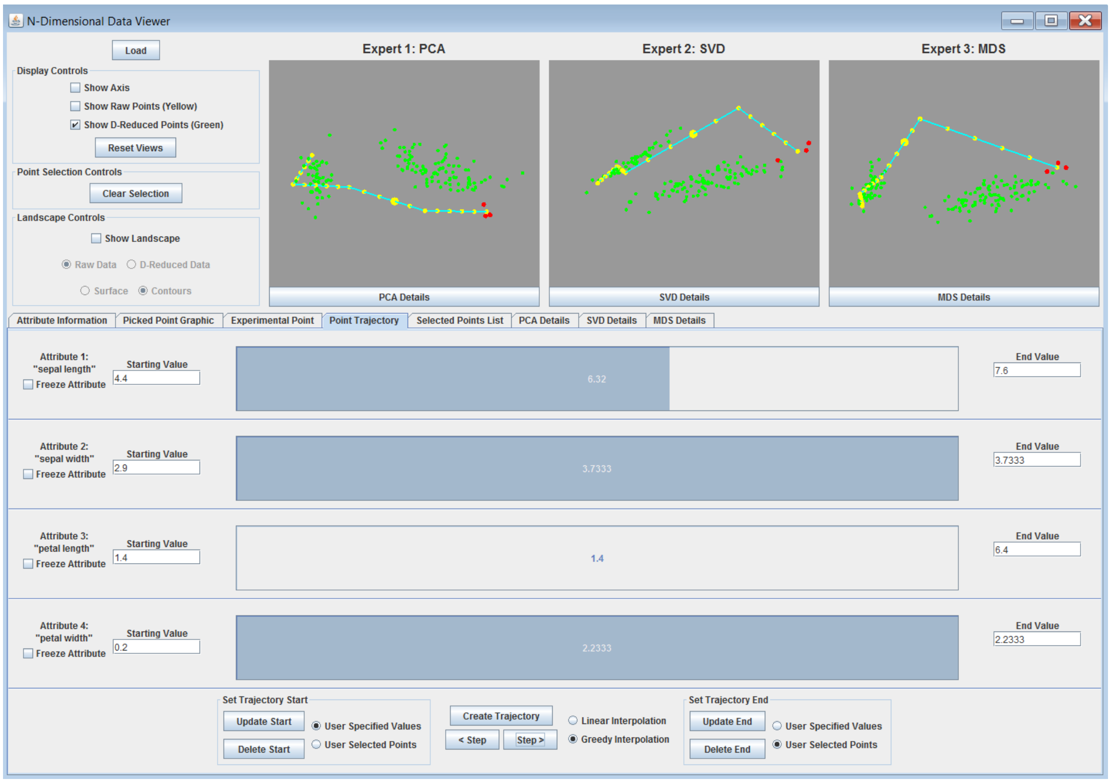
● Linear Interpolation ○ Greedy Interpolation

Set Trajectory End: ○ User Specified Values ● User Selected Points

Figure 4.2: The Point Trajectory user interface. The Iris data is shown in green according to three different dimension reduction techniques, as well as the linear interpolation trajectory. The bottom tab shows the start and end points of the trajectory, as well as the progression of each attribute for the current step of the trajectory.

However, since visual proximity alone does not necessarily guarantee that all attribute values are similar to the surrounding data points, the user can make use of the trajectory generator. Once enabled, the trajectory generator will display a path between the specified start and end positions. The user is then able to advance incrementally along this trajectory until they reach the desired point. The attribute values for the current trajectory step and a progress bar display to the user the progression along the trajectory. Fig. 4.2 shows a partial advancement of along a linear interpolation trajectory between the prior experimental point and a newly desired target position. As previously mentioned, the linearly interpolated trajectory works by dividing the distance between the start and end value of each included attribute from the original data space by an equal number of steps. This results in a straight line trajectory. While easy to visualize, this trajectory may be of limited value if the user finds it difficult to keep track of the changes along each attribute simultaneously.

Instead it may be beneficial for the user to choose the greedy interpolation trajectory. The greedy trajectory is created by interpolating along the attributes one at a time. The attributes are ordered according to the smallest change in value necessary to reach the target value. In other words, the greedy interpolation algorithm selects the attributes of least resistance first, as the first attribute which undergoes modification may be considered the easiest change to make, and so possibly the most desirable.



Display Controls
☐ Show Axis
☐ Show Raw Points (Yellow)
☑ Show D-Reduced Points (Green)

Point Selection Controls

Landscape Controls
☐ Show Landscape
● Raw Data ○ D-Reduced Data
○ Surface ● Contours

Attribute 1: "sepal length" ☐ Freeze Attribute
Starting Value 4.4 — 6.32 — End Value 7.6

Attribute 2: "sepal width" ☐ Freeze Attribute
Starting Value 2.9 — 3.7333 — End Value 3.7333

Attribute 3: "petal length" ☐ Freeze Attribute
Starting Value 1.4 — 1.4 — End Value 6.4

Attribute 4: "petal width" ☐ Freeze Attribute
Starting Value 0.2 — 2.2333 — End Value 2.2333

Set Trajectory Start
● User Specified Values
○ User Selected Points

○ Linear Interpolation
● Greedy Interpolation

Set Trajectory End
○ User Specified Values
● User Selected Points

Figure 4.3: The Point Trajectory user interface. The Iris data is shown in green according to three different dimension reduction techniques, as well as the greedy interpolation trajectory. The bottom tab shows the start and end points of the trajectory, as well as the progression of each attribute for the current step of the trajectory.

Chapter 5

User Feedback

Currently, only limited user feedback has been obtained. Nonetheless, this feedback has been both positive and informative. We selected three users to test the interface and provide feedback two computer programmers and one mathematician. These participants were selected because, though not skilled in the field of data science, their modest familiarity with statistics and data analysis made them good test subjects. The three users were provided the software system and a data set. After a brief introduction, they were each asked to explore the various features of the interface. We requested that the users narrate their thought process as they interacted with the software to help us gain insight into their experience. During each user test, a note-taker recorded the user utterances. In addition, one researcher assumed a coaching role, making suggestions to help guide the novice users upon request or expressions of uncertainty.

We found that the individual manipulation tasks were very easy for the participants to learn. Once the users were shown how to use the particular subsystem, they were clearly able to continue using it without difficulty. In addition, the users appeared confident and motivated to continue exploring the data on their own, and began to make spontaneous deductions and observations about the data. However using the system as a whole by switching between the various tasks was initially difficult and required more coaching. These results seem indicative of the necessity of having an expert user guiding a novice who is learning the system for the first time similar to the previously-described doctor-patient use case. The results also indicate that the system may be particularly useful to novice users who need to perform certain smaller tasks, rather than using the system as a whole.

Also of interest is that the participants extrapolated clear and valid applications for the software in their professional lives. Much of the user feedback was centered around descriptions of how they visualized the software would aid them in performing specific tasks in their daily lives. When asked for potential improvements, the users gave several suggestions. Among these were: a more clearly defined indication of when an attribute within the experimental point exceeds the range present in the original data set; the ability to use different colors to help differentiate different selected groupings; and the ability to set the start or end of a trajectory using the position of the experimental point.

While these initial results are positive, before consideration is given to the release of such an interface, more extensive user studies should be performed in order to verify that this interface has the intended effect.

Chapter 6

Limitations and Future Work

Though the current software implementation does provide the main tools we set out to implement, there are several limitations, as well as improvements which can be made. It should be noted that this software is only intended to be used with continuous numeric data. Nominal data is not supported as it does not lend itself to the dimension reduction or trajectory generation techniques used. Discrete stepwise or ranked data, though usable, will not work well with the current implementation, as the trajectory generation does not currently account for numeric attributes which are representative of stepwise values.

Another issue stems from the nature of some of the dimension reduction techniques used. While PCA produces a set of attribute weights, which can be dynamically applied to a new point, neither SVD or MDS produce models which can be dynamically applied to a newly inserted point. Instead, both of these procedures need to be rerun every time a point is modified. While this brute force method works well for smaller data sets, it can cause noticeable delays when larger data sets are used. Optimization could be made in an effort to mitigate the number of times these dimension reduction techniques need to be run.

Chapter 7

Conclusion

In this paper we have described the design of a software interface intended to provide unfamiliar users the ability to explore high dimensional data. Our intent is that using this interface will enable a novice to develop an actionable intuition for the data in question. The combination of the visualization methods and analysis tools discussed result in a software interface which enables interactive analysis and exploration of high dimensional data. By allowing the user to create and manipulate new data points we enable a novel form of data exploration. With this exploratory point creation, the user is able to directly observe how various attribute values affect the position of the new data point relative to the surrounding data set. In addition, by manipulating the attribute values of this exploratory point, the user can observe the resulting changes in the point's position in real time. These capabilities would be particularly useful when informing a patient of their current health status, based on their unique lifestyle choices, and how various alterations in habits would result in changes to their health. Furthermore, when coupled with the provided interpolation, the user will be shown various trajectories which present the changes necessary to re-position a specified point to a more desirable state. While the motivation behind this system was to aid medical prognostic and diagnostic applications, the ideas discussed are applicable to any high-dimensional data analysis task.

Bibliography

[1] Ankit Agrawal, Sanchit Misra, Ramanathan Narayanan, Lalith Polepeddi, and Alok Choudhary. A lung cancer outcome calculator using ensemble data mining on seer data. In *Proceedings of the Tenth International Workshop on Data Mining in Bioinformatics*, page 5. ACM, 2011.

[2] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear algebra and its applications*, 415(1):20–30, 2006.

[3] UC Davis. Xmdvtool. http://davis.wpi.edu/xmdv. Acessed: 2014-7-1.

[4] Brian S Everitt and Graham Dunn. *Applied multivariate data analysis*, volume 2. Arnold London, 2001.

[5] Joe Faith. Targeted projection pursuit for interactive exploration of high-dimensional data sets. In *IV*, pages 286–292. Citeseer, 2007.

[6] Joe Faith, Robert Mintram, and Maia Angelova. Targeted projection pursuit for visualizing gene expression data classifications. *Bioinformatics*, 22(21):2667–2673, 2006.

[7] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[8] Imola K Fodor. A survey of dimension reduction techniques, 2002.

[9] The GGobi Foundation. Ggobi. www.ggobi.org. Acessed: 2014-7-1.

[10] Jerome H Friedman and John W Tukey. A projection pursuit algorithm for exploratory data analysis. 1973.

[11] Rocio Garcia-Retamero, Yasmina Okan, and Edward T Cokely. Using visual aids to improve communication of risks about health: a review. *The Scientific World Journal*, 2012, 2012.

[12] Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.

[13] Harry H Harman. Modern factor analysis. 1960.

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

[15] Max A Little, Patrick E McSharry, Stephen J Roberts, Declan AE Costello, Irene M Moroz, et al. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMedical Engineering OnLine*, 6(1):23, 2007.

[16] Microsoft. Microsoft business analytics. http://www.microsoft.com/en-us/server-cloud/audience/business-analytics.aspx. Acessed: 2014-7-1.

[17] Eun Ju Nam, Yiping Han, Klaus Mueller, Alla Zelenyuk, and Dan Imre. Clustersculptor: A visual analytics tool for high-dimensional data. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 75–82. IEEE, 2007.

[18] A Michael Noll. A computer technique for displaying n-dimensional hyperobjects. *Communications of the ACM*, 10(8):469–473, 1967.

[19] Yukihito Sakai and Shuji Hashimoto. Four-dimensional mathematical data visualization via embodied four-dimensional space display system. *Forma*, 26(1):11–18, 2011.

[20] Tableau Software. Tableau. www.tableausoftware.com. Acessed: 2014-7-1.

[21] R Steven. *Hollasch:Four-Space Visualization of 4D Objects,*. PhD thesis, MS Dissertation, Arizona State University (August 1991), 1991.