A DATA VISUALIZATION TOOL TO IDENTIFY PATTERNS FORMED BY SUBSETS OF DATA

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Md Golam Morshed Osmani

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

April 2010

Fargo, North Dakota

North Dakota State University
Graduate School

Title

A Data Visualization Tool To Identify Patterns Formed By Subsets Of Data

By

Golam Morshed Osmani

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

ABSTRACT

Osmani, Md Golam Morshed, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, April 2010. A Data Visualization Tool to Identify Patterns Formed by Subsets of Data. Major Professor: Dr. Anne Denton.

An object may be identified by its properties which may comprise both continuous data values and categorical data values. Sometimes a particular property (categorical) value may select a group of objects which have similarity in their attributes (continuous). This feature is observed in gene data sets as well as other multivariate data sets.

This paper presents a tool to identify patterns formed by data subsets based upon some categorical attributes. The tool reads data from various data sources, filters data according to supplied criteria and shows them as a Parallel Coordinates graph which a user can manipulate to find the relation between data subsets by changing the selection criteria. The user is given a choice to change values of the various categorical columns and to select a subset of data to find out their similarity.

This tool can be used both in desktop and web environments. The class library is implemented in C# and also ported to Java to make it widely available. Other application developers can also use this library in their application to have this functionality readily available.

ACKNOWLEDGEMENTS

I am grateful for the help and inspiration I got from all the faculty of the Computer Science Department. I am especially grateful to Dr. Anne Denton, who is an excellent mentor, for her assistance in every aspect of this paper, her support, her guidance and her motivation. I would also like to thank my supervisory committee members for their excellent support. Without their help, it would have been impossible for me to come up with this paper.

I would also like to thank to my parents and my beloved wife for continuous support in my study life.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

CHAPTER 1. INTRODUCTION

In machine learning and applied statistics, objects under consideration are often assumed to have attributes which identify and describe the objects, so finding the relationship between the objects can imply the relationship between corresponding attributes. The objects of interest could have continuous as well as discrete/categorical attributes. Sometimes these discrete/categorical attributes may identify a particular group of objects which may have some degree of similarity between them in terms of continuous attributes. Thus a trend in data could be seen for those grouped objects. For example, in bioinformatics, the attribute set of protein domains in a gene can consist of both continuous and discrete/categorical values. Continuous attributes include the time series gene expression data, while discrete attributes include the properties "participation in a function," "cause of a disease," etc. Sometimes, protein domains of different genes sharing the same property (like participation in the same/a similar function) may have similar gene expression profiles, which can be easily noticed visually by making these profiles more prominent than others. Thus, filtering a gene subset which may have a certain property in common and visually isolating the subset from others could led to patterns which can be characterized by the presence of a similar gene expression profile. These patterns may be very useful in further study of bioinformatics. The same could be true for other multivariate data sets which may have both continuous and discrete/categorical attributes.

For a positively correlated, two dimensional (2D) data set, if both dimensions are plotted along the Y axis, two vertical lines are used, one for each dimension. The

values in those two dimensions are marked on those two lines, and the connecting line represents the point. This process can be continued arbitrarily for many dimensions. Figure 1 shows a simplified version of this scenario or a negatively correlated 2D data set, lines tend to cross each other, which is shown in another simplified figure (Figure 2).

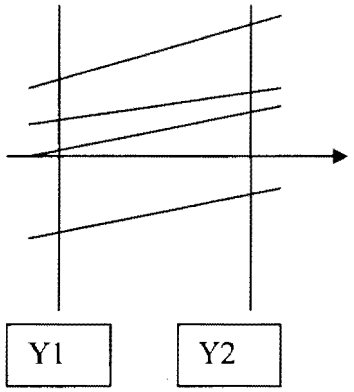

Figure 1. Positively correlated two-dimensional data set.

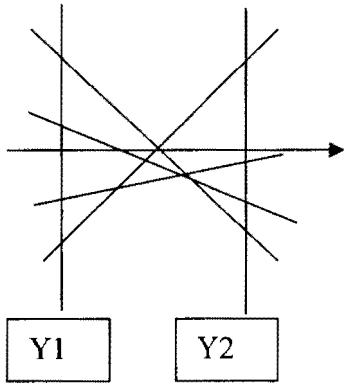

Figure 2. Negatively correlated two-dimensional data set.

Both positive correlation and negative correlation indicate the presence of a pattern. Sometimes axes could be uncorrelated when the lines neither clearly look parallel, nor cross each other; rather, randomly distributed. An example of an uncorrelated data set is given in Figure 3. Thus, this idea of having positively or

negatively correlated dimensions as pattern can be extended for a high dimensional data set, where carefully chosen dimension ordering in Parallel Coordinates [1] can relate the correlation between consecutive dimensions to the overall relationship between them.

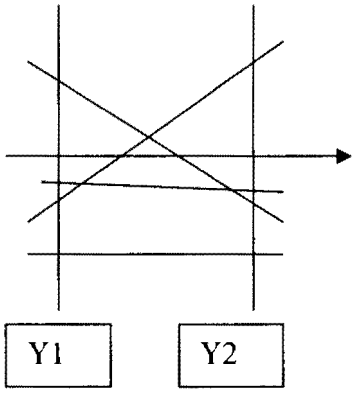

Figure 3. No correlation in two-dimensional data set.

Parallel Coordinates represent a multidimensional data set as two dimensional line segments where each line segment represents a point in the multidimensional hyper plane. Proximity between these points in the hyper plane can be translated as the distance between the corresponding line segments in the Parallel Coordinates system. Thus, the similarity measurement in a two-dimensional Parallel Coordinates system has the corresponding measurement in the multidimensional hyper plane. For example, block distance, Euclidian distance, cosine vector, etc. can easily be measured in a Parallel Coordinates system while the effect of normalization is considered. If the values are normalized and they are from related attributes or dimensions, another measurement can be used to find similarity between data items. A Pearson's correlation coefficient can be measured between data items as a measurement of similarity between them.

1.1. Problem Statement

There are several tools being used for data visualization of various types of data sets. Some tools work with gene expression data sets, some with time series, and some with any type. These tools can represent the data in various forms, like graphs and plots. These tools are good in their respective area. However, experience is required to work with these tools. Also, simple manipulation in the data may need some predefined steps to follow, with which a new user may feel uncomfortable. The solution is a simple tool which would be easy to use, have zero learning time, and would be easy to manipulate.

The primary objective of this research study is to develop a data representational tool to identify patterns among data subsets based upon some categorical attribute values. The tool should be able to filter the data set by discrete or categorical attributes and show the continuous data using Parallel Coordinates, thus helping to identify the objects of interest. This tool can be used as a library as well as a standalone application in both desktop and web environments. The user will select the values of the categorical attributes, and based on the selection, filtered data items will be shown on the graph.

1.2. Organization

The paper is organized as follows.

Chapter 2 includes the Background Study of the related work.

Chapter 3 contains concepts and the Current Solution.

Chapter 4 describes how to use this tool.

Chapter 5 provides the Results.

Chapter 6 provides the Conclusion.

CHAPTER 2. BACKGROUND STUDY

This chapter provides an overview of the related works in this area. A more detailed discussion of these areas can be found in the literature referenced through this chapter.

2.1. Related Work

There are several commercial and open source solutions available for generating charts and graphs from a data set. Most of them are desktop solutions although some web solutions exist. They are good in displaying data or a trend in the data for day-to-day activities. Several tools have been developed to analyze gene expression data using pathways. Most of the tools determine the characteristics of differentially expressed genes by using overlap statistics such as the cumulative hyper geometric distribution [2]. In 2005, Subramanian et al. [3] presented a study on interpreting gene expression data using Gene Set Enrichment Analysis (GSEA) tool. The tool focused on gene sets such as groups of genes that share a common biological function, chromosomal location, or regulation, and revealed many biological pathways in common.

Denton et al. [4] reported a research study about developing an algorithm to relate the patterns of gene expression in a set of microarray experiments to a functional group, such as protein function, in one step. The basic assumption of this research study was that patterns co-occur frequently. Density histograms were developed using product similarity among expression vectors and were used to evaluate the relationship between expression data and functional annotations. This approach also tried to overcome the limitation of GSEA. Two basic objectives of this research algorithm were

to identify subsets which were significantly different from what would be expected for a random subset and to search for data points that had more neighbors than expected. This study performed biological analysis of functional groups of proteins, developed hypotheses for future biological studies, and tested one hypothesis experimentally. The researchers confirmed that the theoretical model can achieve the scaling of the algorithm with a large data set.

Denton and Wu [5] as well as others in the data mining and machine learning community postulated one of the most influential concepts in data mining which considers multiple, continuous attributes as dimensions in a vector space. The research study considered the continuous attributes as vector data and investigated the patterns that relate vector attributes to one or more subsets. The competence of the proposed approach was evaluated by cell-cycle gene expression data, time series subsequence data, and coating data. The research study conjugated the continuous vector data and item data which are enormously important in many areas of application.

In 2009, Charaniya et al. [6] presented a study considering three complementary tools for gene set testing (GST) to identify the pathways of high-producing cell lines as well as the biological functions which were significantly altered in high producers. Using the GST tool, the research study was able to identify groups of functionally coherent genes. These tools were employed to discern the statistical significance at a functional level rather than at the individual gene level. This study demonstrated the values of using GST as a complementary tool to analyze the gene-level differential expression.

A number of works have been done with clustering gene expression data to find the genes that show a similar differential expression pattern under predefined conditions [7, 8, 9]. However, these research studies did not directly relate gene expression to domain or functional information while functional information has a significant effect to improve the clustering results [10, 11]. Identification of functional groups and the genes that belong to them in a single step can be achieved through the biclustering technique [12]. Gene differential expression in time course experiments had also been identified using expression patterns.

2.1.1. Related Work in Data Visualization

Several commercial and non-commercial (open source) software/libraries are available for data visualization. Valentini [13] developed the Mosclust library to discover an interesting pattern in data by statistical computation and visualization. Merja Oja et al. [14] used a MAP based technique (Median Self-Organizing Map) to group and visualize some patterns (human endogenous retroviruses).

WEKA [15] is an integrated software which contains tools for data pre-processing, classification, regression, clustering, association rule mining, and visualization. The user can either use WEKA as standalone software or can use these algorithms in his or her Java program. The benefits of WEKA are that it is easy to use, has a huge choice of algorithms, manipulation on the input data, etc.

Shneiderman et al. [16] used TimeSearcher to represent time series data. TimeSearcher can be used to identify and search pattern among time series. It gives the user the ability to interact with data and patterns.

CHAPTER 3. CURRENT SOLUTION

We need a tool to identify the relationship between genes based on a particular function or property (which may be either binary or categorical). Other commercial and non-commercial tools can perform the same task but need some data processing prior to getting the result. This data preprocessing may often lead to unnecessary complexity when the requirement is only to identify a similar pattern/trend among data in a continuous data set based on a particular choice of categorical value.

A macro/Visual Basic for Application (VBA) code in MS Excel can also be used for this purpose because MS Excel has a rich set of graphing tools which can be used to see trends in data. However, MS Excel is not free and may not be available to all the interested audience members.

My solution is to develop a customized tool, both as standalone application and as a reusable library. This tool can be accessed online, so people may not need extra software, just a web browser and an internet connection.

3.1. Concepts

There are several ways to represent data visually. Most of them work well with data that have few dimensions, but a multivariate data set can have a large number of dimensions and may not be easy to represent in the existing visualization systems. The common graphs (scatter graph, line graph, bar chart, etc) are good for a two / three dimensional data set. For an example, we could have a scatter plot of highway-mpg and

price attributes for an automobile data set [17]. In Figure 4 highway-mpg is mapped across x-axis, and price is in y-axis.

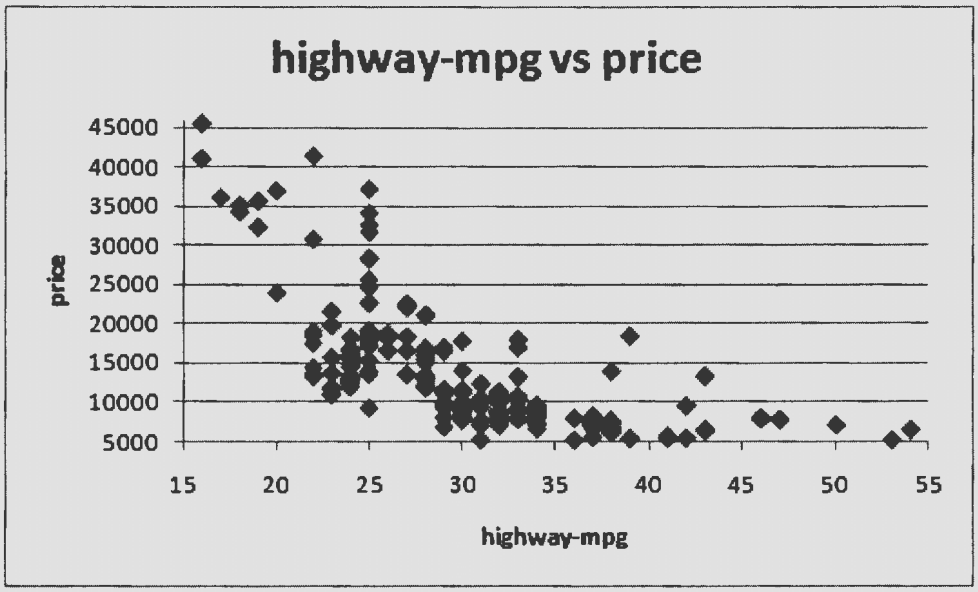

Figure 4. Scatter plot of highway-mpg and price (an automobile data set [17]).

Even though our visualization systems (computer monitor, screen projector, etc.) are two dimensional, we could also plot three-dimensional data using projection on the screen. Beyond that, it becomes very difficult and unappealing to draw a multidimensional graph on the screen using projection.

Several solutions for the multivariate data visualization have been proposed by researchers. These solutions can be categorized as the "axes reconfiguration technique," (such as Parallel Coordinates and glyphs), "dimensional embedding system" (worlds within worlds, dimensional stacking, etc.), dimensional reduction techniques, etc. [1, 18]. Sometimes, scatter plots are used, and in this case we get n*(n-1)/2 scatter plots, where n is the number of attributes in the data set.

In my solution, I have used the Parallel Coordinates technique which is a very useful and efficient technique to visualize a multivariate, large data set in a 2D environment. The idea behind Parallel Coordinates is very simple. Attributes are represented as vertical lines (columns), and the connecting line between the corresponding values in all the vertical lines represents a multidimensional point. Thus, each connecting line corresponds to a data point in the data set. The Parallel Coordinates technique can be seen as a generalized version of the 2D Cartesian coordinate system. Instead of having N axes in separate directions, all the axes are drawn in parallel (vertically). Here, duality is observed; the dual points become lines, and dual lines become points.

The most important use of the Parallel Coordinates is exploratory data analysis (EDA) [1] for discovering data subset relationship. For a data set with N items, the subset could be any of the 2^N possible subsets. Thus, it may not be feasible to run an exhaustive search as 2^N is not a polynomial but an exponential function. Human eye can find some good patterns and can identify them. The Parallel Coordinates transform multivariate relationships/patterns into 2D patterns. Thus, searching for patterns in a multidimensional data set becomes simple problem of finding 2D patterns that are readily perceivable by the human eye.

The main advantage of the Parallel Coordinates is that the number of dimensions it can support is only limited by the horizontal resolution of the display device. Closer axes reduce the visibility and it may be hard to get any meaningful pattern. Another interesting feature of Parallel Coordinates is that correlation between attributes can be easily identified. This technique works well when the attributes are ordered in some

way; related and relevant attributes are placed together. Otherwise a cluttered data set may not give us any meaningful insight. Another challenge is the high volume of data. Visualization of a large data set may cause loss of speed, overlapping objects, and a reduction in readability. Objects (lines) are piled up one after another, and some of them may not be visible at all and fall just behind others. This complexity often reduces the amount of information the graph could present.

The Parallel Coordinates can also provide statistical interpretation of data. For example parallel (or almost parallel) lines between axes show positive correlations between them, whereas crossing lines between parallel axes denotes a negative correlation between the attributes. Figure 5 shows the positive correlation between city-mpg and highway-mpg whereas Figure 6 shows the negative correlation between normalized loss and wheel-base (an automobile data set [17]).

Data normalization is another point which must be addressed in this tool. Each attribute (represented by each vertical line) has its own range which may not be compatible with others. For example "height" may have different range (47.8-59.8) than "price" (5118-45400). Thus, normalization is required to place the data points on the graph. I have used two types of normalization, range normalization and statistical normalization. Range normalization provides a simple solution and can show the data trend, but is highly affected by extreme values. An unexpectedly high number can not only affect the axis, but also affect the lines between the current axis and adjacent axes. Statistical normalization, on the other hand, is less affected by the extreme values. Though the current axis is still affected by extreme value, lines between current and

adjacent axes are not much affected except the lines for extreme values. Statistical normalization also points to possible error data values.

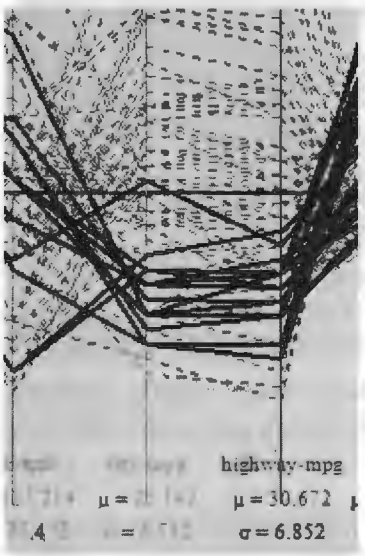

Figure 5. Positive correlation.

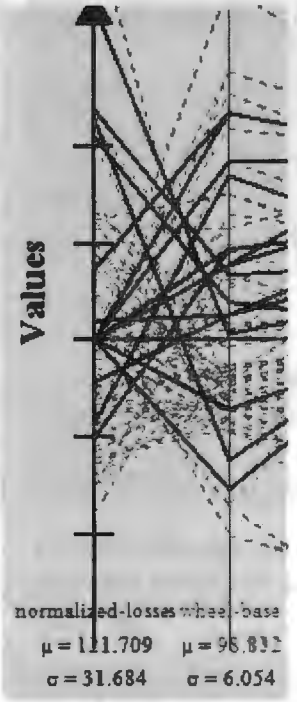

Figure 6. Negative correlation.

Figure 7 shows a portion of the graph using selection criteria (num-of-cylinders="six" and risk-type=6) on the automobile [17] data set. Here, the range normalization technique is used. Figure 8 shows the same graph, but the data set has one error value: 3.35 becomes 335. Figures 9 and 10 shows the same graph and respective data sets, but the normalization technique has been changed to statistical normalization. From the figures (7, 8, 9 and 10) it is very clear that extreme values can be easily identified by statistical normalization.

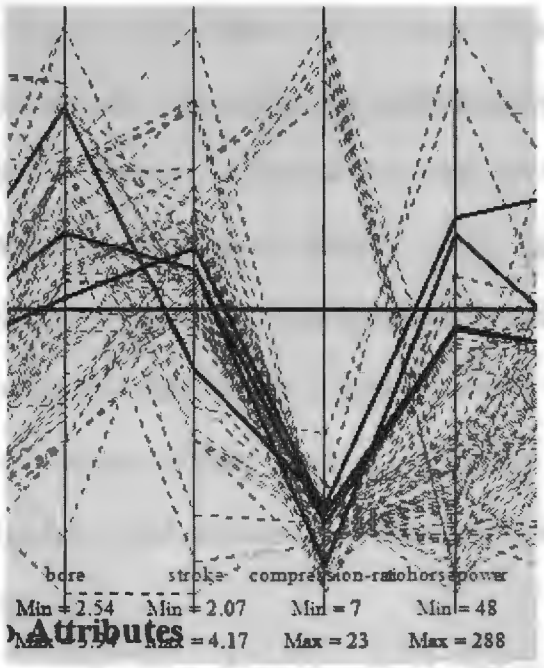

Figure7. Range normalization with no extreme value (an automobile data set [17]).

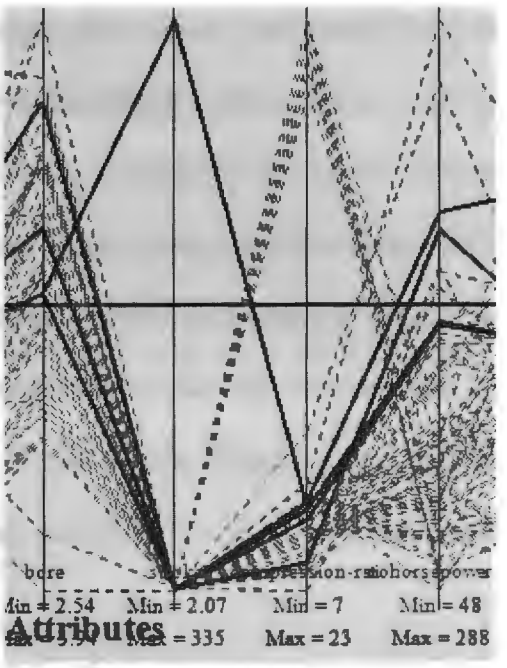

Figure8. Range normalization with extreme value (an automobile data set [17]).

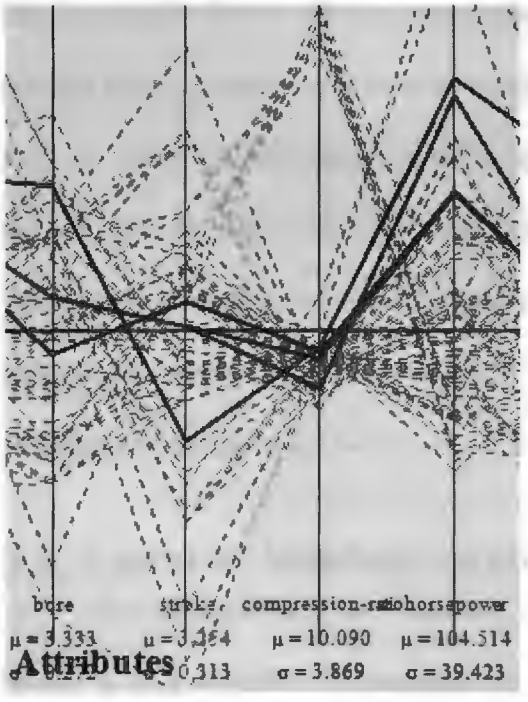

Figure9. Statistical normalization with no extreme value (an automobile data set [17]).

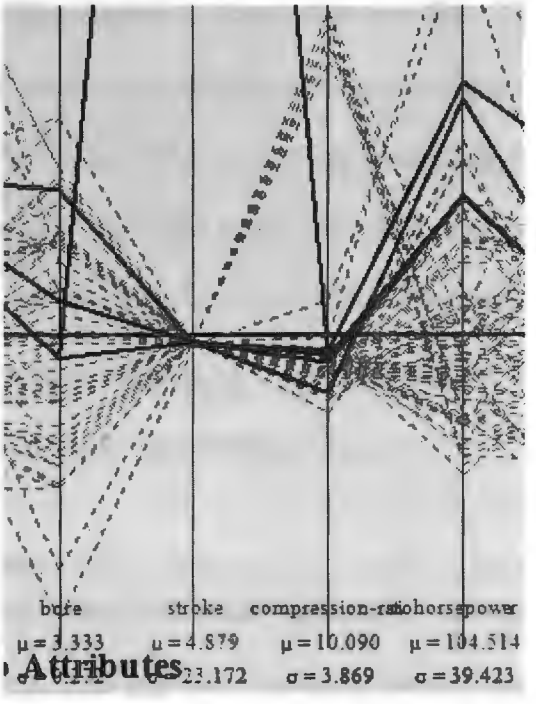

Figure10. Statistical normalization with extreme value (an automobile data set [17]).

A single attribute or a group of categorical attributes can identify a group of objects where the continuous attributes have similar trends in their values. For example, let us consider a hypothetical data set where a person's six forms of personal information are given: height, weight, age, income, race, and education. Of them, height, weight, age, and income are continuous data, and race and education are categorical data. Height is given in inches, weight in pounds, age in years, and income in thousand dollars. Race could be any of the values (White, Asian, Black, and Hispanic), and education is in high-school and graduate. This is an artificially generated data set.

A subset of data identified by Race=Asian shows a pattern in the selected subset. These data items are presented as thick lines and they show a trend when all of them simultaneously rise or fall for a particular axis. Figure 11 shows this selection. These persons show similar trends in their height, weight, age and income, so the categorical attribute value "Asian" separates a group of people whose other attributes follows a pattern. On the other hand, some choices may not yield much similarity. Figure 12 shows the subset of data where all persons are "Hispanic" by race. This subset does not show many similar trends or patterns. (The data set used in Figures 11 and 12 is an artificial one and does not necessarily reflect real life observation.)

3.2. Current Implementation

This tool comes in two versions, desktop and web based application. A reusable library is also available to integrate this tool into any other program. If this tool is accessed through the web, it may have subset of functionalities. However, a web

application developer can take full advantages of this solution. This solution is primarily developed in C#. However, a Java port is also available. Web solutions are available both as C#/ASP.NET application and as Java applet. Desktop solutions can both work with a text file and a relational database. A web application user can follow the text-file approach, but the web application developer can also enjoy database support.

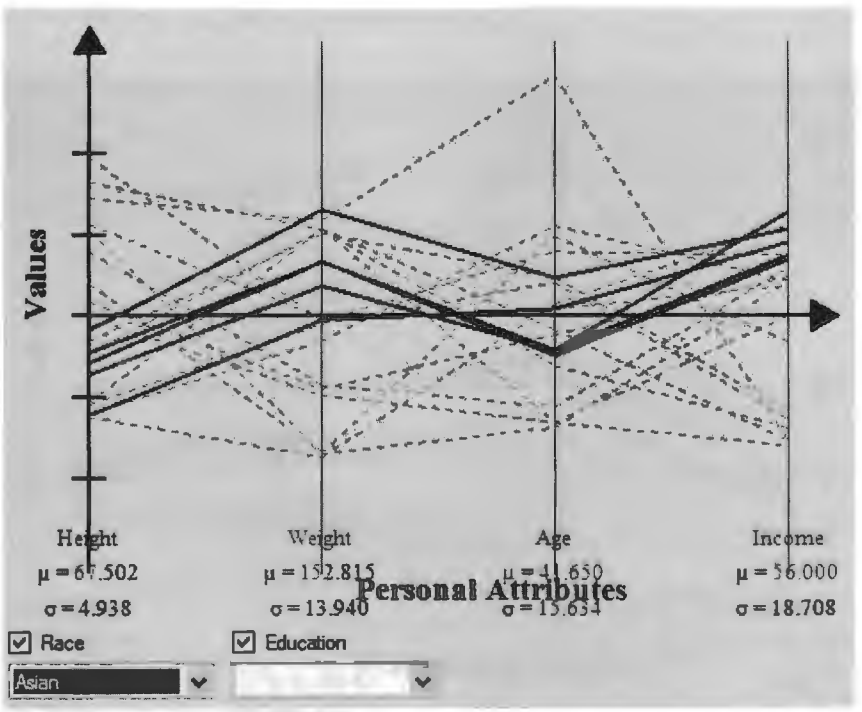

Figure 11. Similar trend in continuous attributes (Race = Asian).

With a desktop solution, a user connects to a data source which may be either a pure text file in a predefined format or a relational database for which some extra parameter values are to be set after the connection parameter is set. The text file would contain i) caption of the graph, ii) X-axis label, iii) Y-axis label, iv) types of data columns, v) labels for data columns and vi) data rows. A sample data file containing

data in the proper format and with the proper labels is shown in Appendix A. A relational database can also be used as a data source by providing the connection string which would contain the database source and connection parameters. The format and a sample connection string can be found in Appendix B. A third hybrid approach is used when the entire database configuration is stored in a text file but the actual data are read from a database using the metadata info from the configuration file.

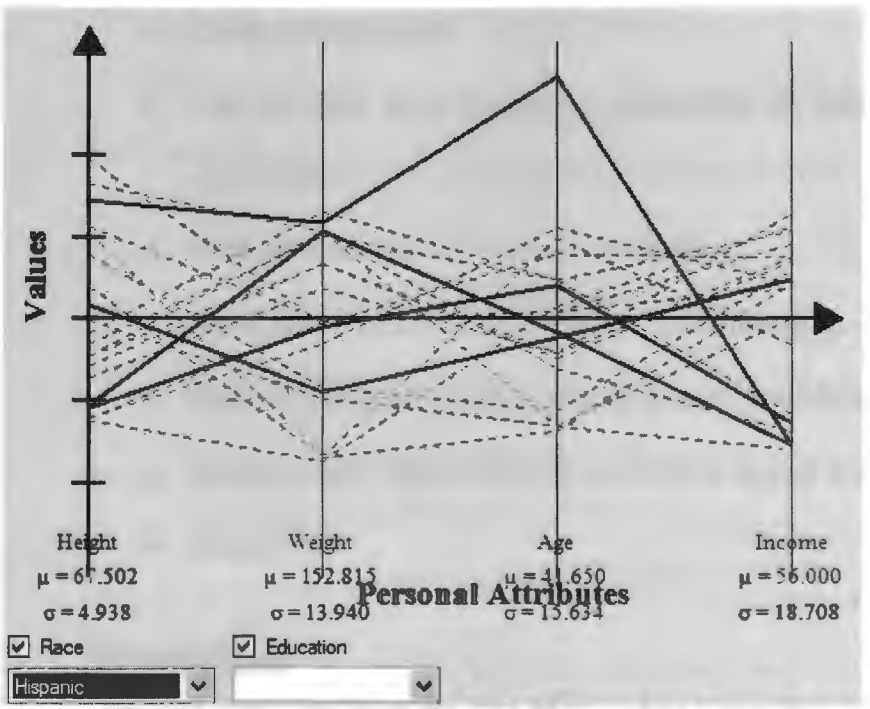

Figure 12. No clear trend in continuous attributes (Race = Hispanic).

A web solution may have two approaches. A simple web application user can copy and paste the data from the text file and also fill the other parameter values. Then, he or she submits the form, and the data are displayed as an image. The user can interact with the form and see the new pattern. In the developer mode, a developer can use this

tool to read from a text file or a relational data source, and then display the graph as an image. External users can also interact with this graph, and they could be given the choice to load data from multiple data sources that are available.

The web version comes in two different versions: ASP.NET application and Java applet. The user can either utilize the web control in his/her ASP.NET application or embed this Java applet in any type of web page, even a static HTML page.

Some advantages of the tool are

- Easily customizable
- Can be used as a standalone application or integrated into another application
- Both desktop and web versions available
- Open source (freely modifiable and redistributable)
- Can add additional functionality to fit user requirements
- Ported to Java which makes it available in almost every environment
- Easy to use

3.3. Workflow

The tool reads data from the data source, does some processing on the data, and displays the graph on the canvas. This canvas could be a control or a form in desktop mode, and an image in web application. The user interacts with this tool and changes the categorical attributes values. The graph changes each time the user changes the selection of categorical values. The user can change the data source any time, and the new graph is displayed immediately after he or she changes the data source.

3.3.1. Data Processing

Data processing is required before the data is used in this tool. This tool does not recognize a missing data entry in any column. Rows with missing data in categorical columns are removed before the data is used, and for numeric columns, the average value substitutes for the missing value.

3.3.2. Data Read and Graph Creation

The followings steps are used in graph creation:

- The user selects a data source (a file or a database or a configuration file)
- Data are read from the source. For a plain text file, all the metadata and graph data are stored in the same file in a predefined way (Appendix A). For a database, all metadata are collected from a wizard, and data are read from the database (Appendix B). A third way is to read metadata from a configuration file (Appendix C) and graph data are fetched from the database as specified in the configuration file.
- Retrieved data are stored in a custom local data class which holds the data, has the column metadata information and calculates other statistics for the numeric column (maximum, minimum, average, and standard deviation). For string data, the lists of all possible values are extracted. The same is true for integer columns unless they are marked as non-categorical column. A normalized data set is calculated from the read data set which will be used in drawing the graph.

- The graph is drawn each time the container refreshes itself. Values on the Y-axis are capped in between 3σ and -3σ where σ is the standard deviation for the column.
- The container of the graph is responsible for creating the combo checkboxes as well as checkboxes for each categorical column. Combo checkboxes are filled with the list of possible values that the column can take.
- The user selects some criteria; the graph is changed based on the selected criteria. Only matching lines are shown in bold solid lines whereas all other lines are shown in dashed thin lines.
- At any time the user can load another data set from any other source meeting the data source criterion.

3.3.3. Schematic Diagram

The above operations can be shown in a following schematic diagram (Figure 13).

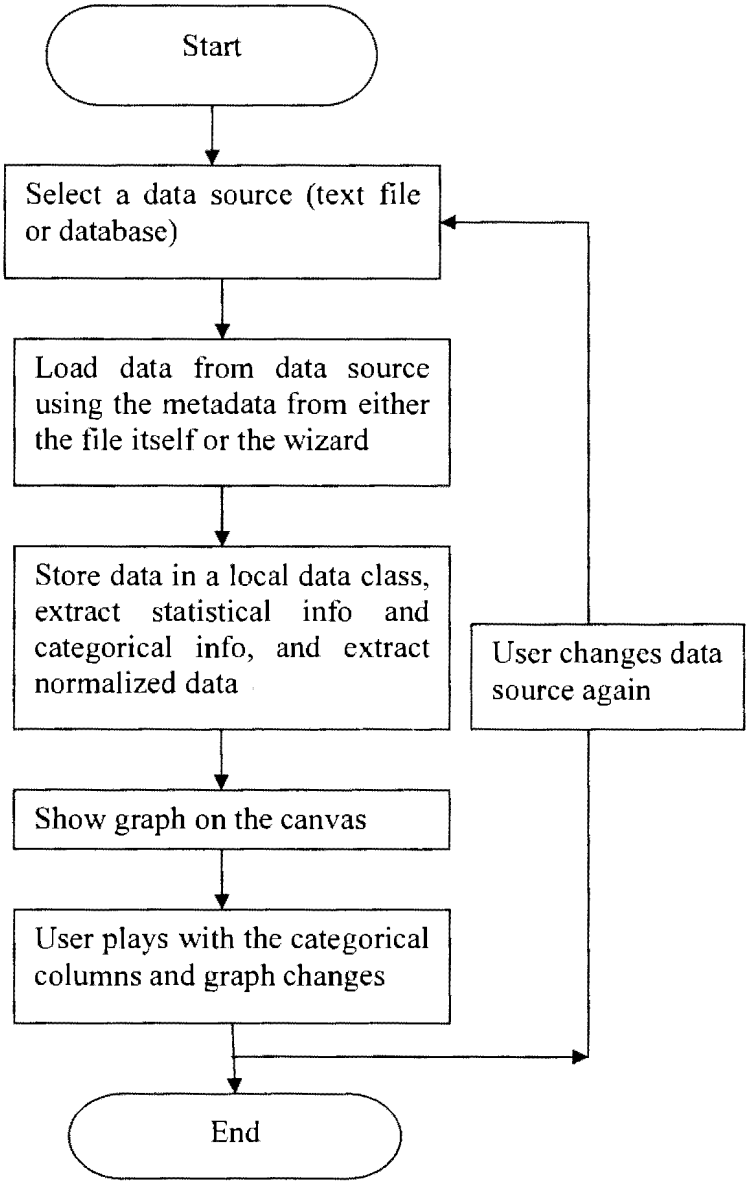

Figure 13. Flowchart of operation

CHAPTER 4. USAGE OF THE TOOL

4.1. Organization

This chapter provides an overview of the tool and how to use it. It describes the two primary ways to work with this tool, a desktop and the web. The desktop part can read data from two sources, a file and a database. The web part can also do the same but may need little more support to show the web application user various ways of loading the data to the tool; for example, data can be fed from a web form input box or file upload control, from a list of previously uploaded files, or from a list of available tables from a supplied data source. The desktop tool is fully implemented here, whereas the web part is presented here to show the user some possible ways to use this tool on the web.

4.2. Desktop Tool

The desktop tool can be used in two modes, file mode and database mode. In file mode, a user/programmer assigns a text file in a predefined format to this tool; the tool reads data from the file and displays it. Then, the user can interact with the graph, can customize it, and can even reassign another file and see the graph from the new file. In database mode, the user puts the connection string in the input dialog. Then, the tool reads metadata from the database, and shows the user a possible list of continuous data columns and a choice to include them in the graph. After that, the user has the choice to include categorical/binary columns which would decide which subset is to be shown as thick lines on the graph. A third hybrid mode exists where database configuration is

stored in a configuration file and actual data are read from a database using those metadata.

4.2.1. File Mode

The user loads a file from a local file system. Figure 14 shows the "Data" menu from where the user chooses the file data source. Figure 15 shows the file open common dialog which comes after selecting the "From File" menu item. Then the graph is shown on the canvas. Figure 16 shows the screen just after the data are loaded from files. In this state, the graph does not have any selected lines (data sub set).

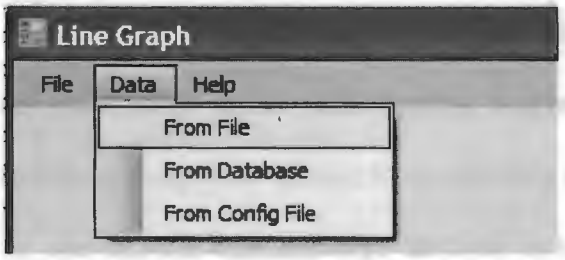

Figure 14. Data menu selecting "File" as the data source.

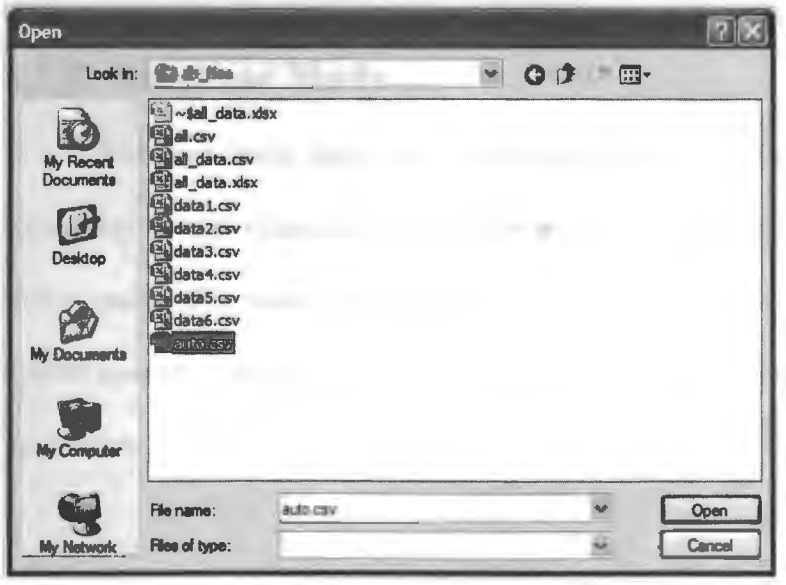

Figure 15. Selecting input file.

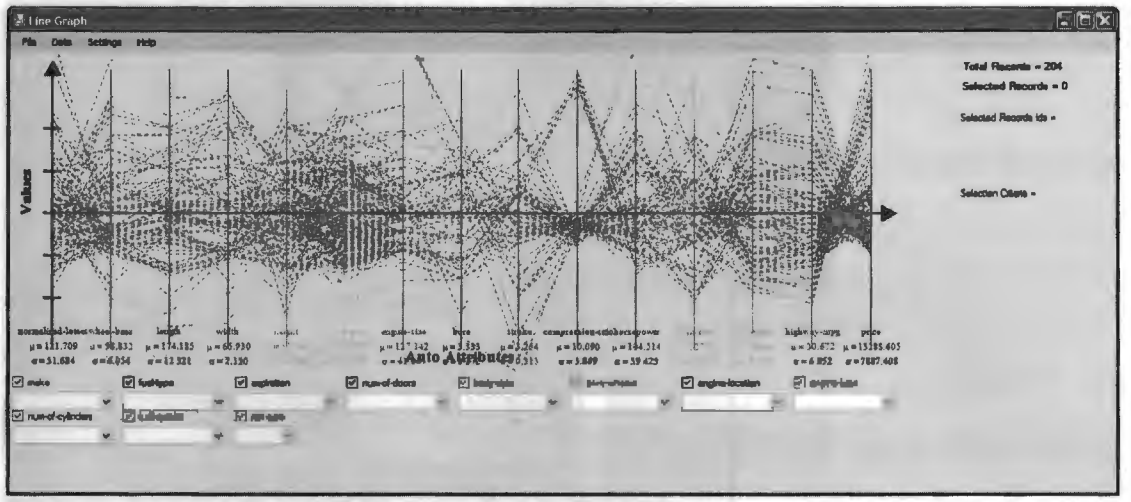

Figure 16. Graph from "File" data source

The user can change the selection criteria by choosing some of the items (which are checkboxes) in a combo box by clicking on them. The user can also completely remove a categorical column by unselecting the top checkbox for that column. In Figure 17 one can see that fuel-type attribute is totally unselected and that the number-of-cylinder attribute is being selected to "six".

4.2.2. Database Mode

The user loads data from a relational database management system (DBMS) which can be any relational DBMS that support open database connectivity (ODBC). Before using this mode, the user may need to create a data source name (DSN) in his/her system. Figure 18 shows the menu when the user selects the "From Database" menu item.

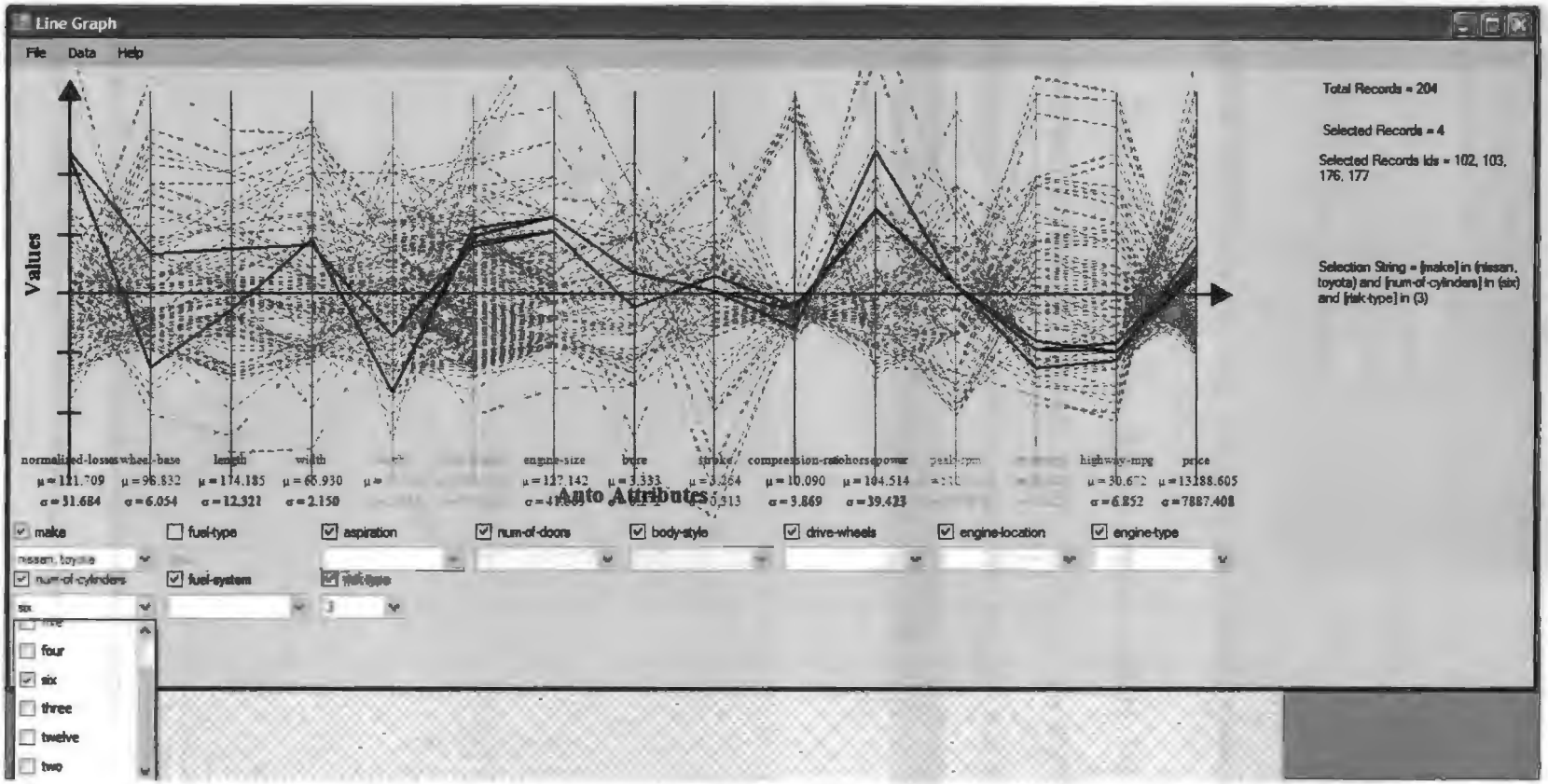

Figure 17. Graph manipulation, attribute value changing.

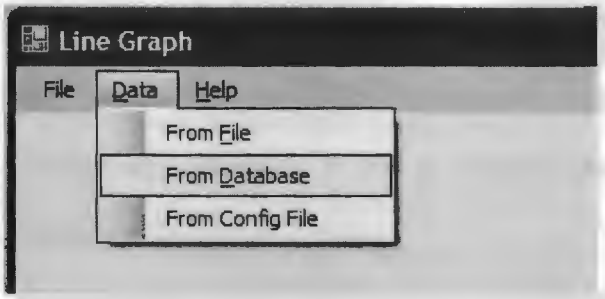

Figure 18. Data menu selecting "Database" as the data source.

Then, the Data Selection Wizard appears (Figure 19). This is the welcome screen of the Data Selection Wizard which will led the user to select the database, table, continuous and categorical attributes of that table, legends, and graph caption.

Figure 19. Data Selection Wizard welcome screen.

When the user continues from Figure 19, he or she gets the connection string entry page where an ODBC connection string is entered to connect to a database. (Please see Appendix B for a sample connection string). Figure 20 shows the connection string entry dialog.

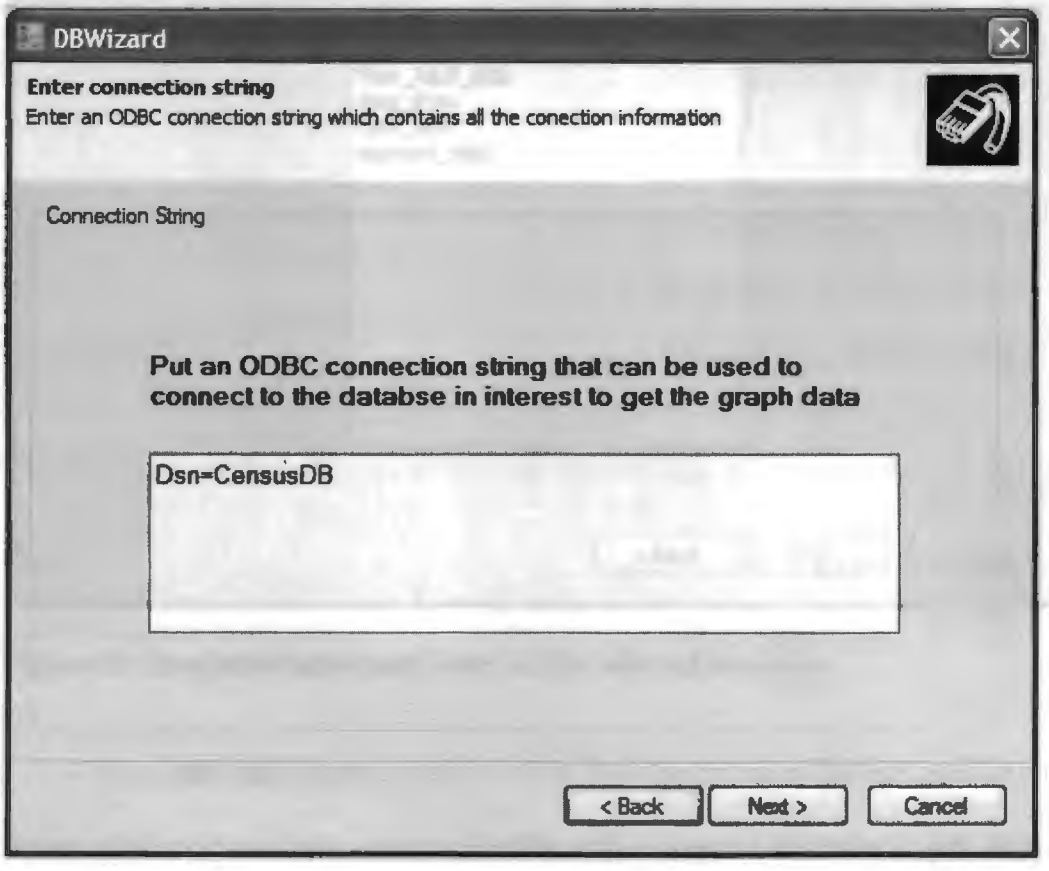

Figure 20. Database Connection string entry.

Available tables in the database are shown when the user enters a valid ODBC connection string and hits the "Next" button. If the user enters an invalid connection string, he or she stays in the connection string entry dialog and an error message is shown. In Figure 21, all the tables and views which are present in the database pointed to by the connection string entered in the previous dialog are shown.

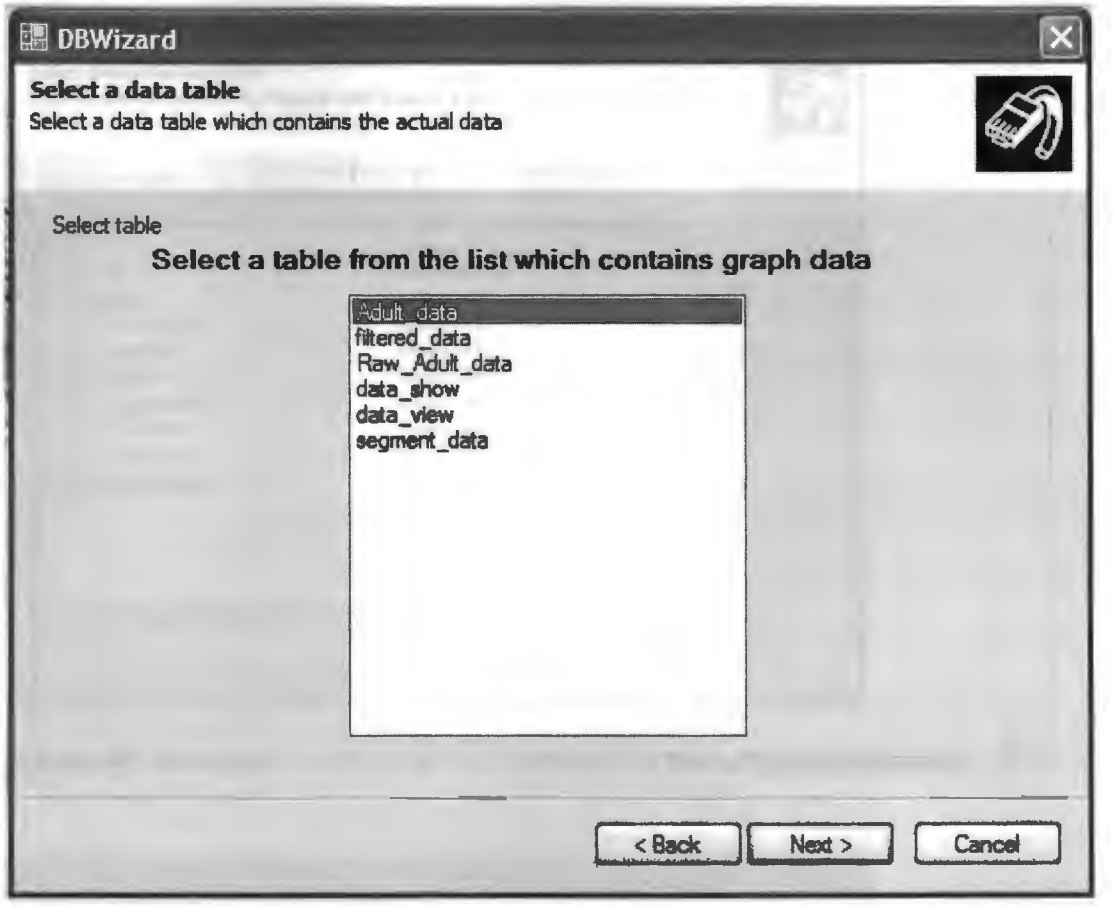

Figure 21. Available tables and views in the selected database.

Now, the user selects a table or view from the previous list and goes to the next screen which shows all the numeric (double and/or integer) attributes in that table/view. The user can select all of them or some of them. These attributes will serve as continuous attributes. Figure 22 depicts the dialog where the user selects continuous attributes (data columns).

After selecting data attributes, the user goes to the next screen where he or she selects the categorical attributes which would be used to filter the data to produce subsets. The categorical attributes' selection is shown in Figure 23.

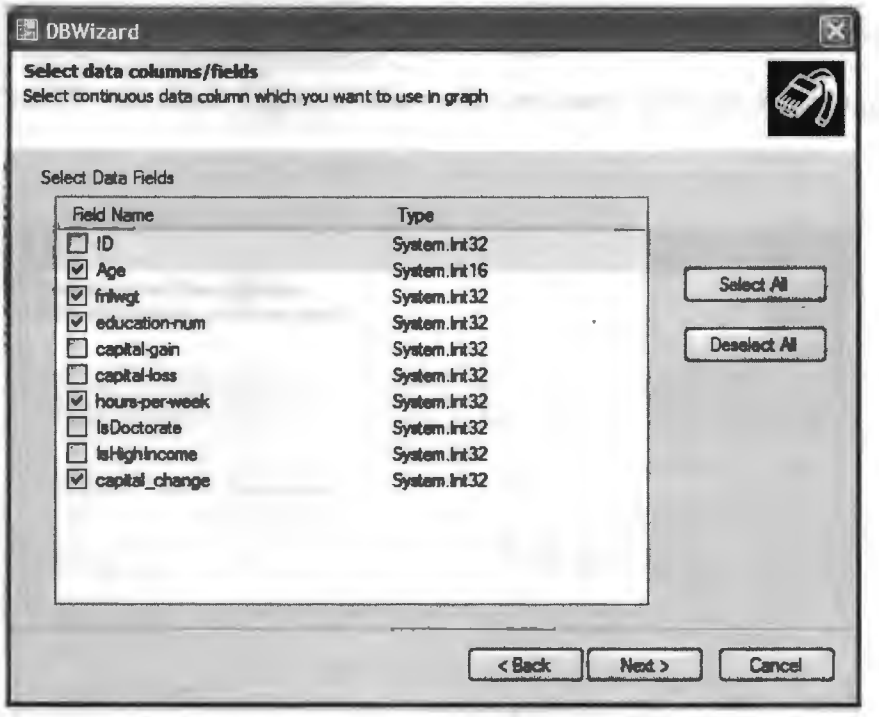

Figure 22. Available continuous data columns in the selected table/view.

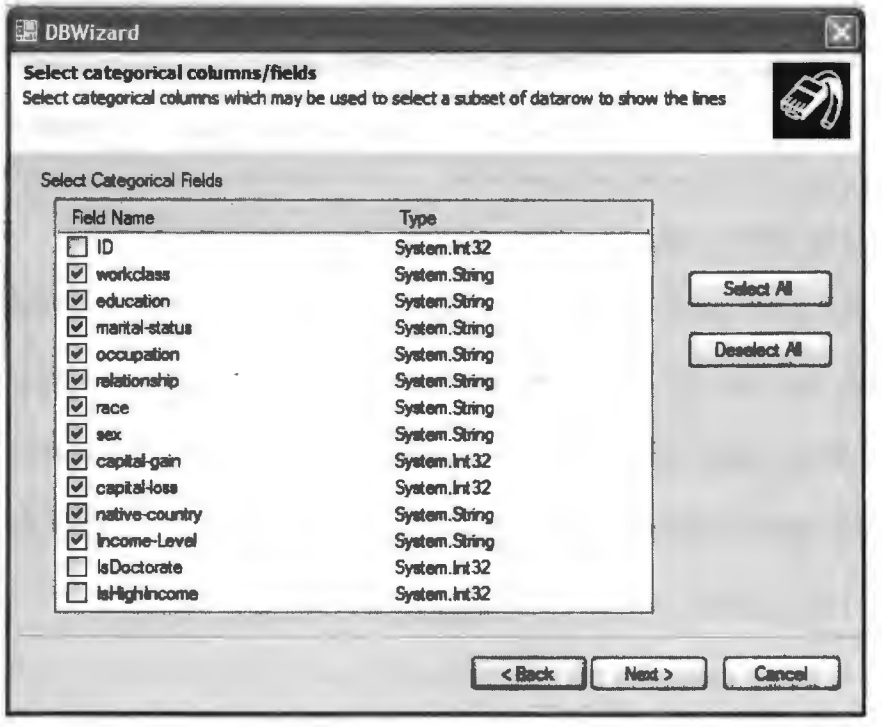

Figure 23. Available categorical data columns in the selected table/view.

Then the user provides the legends which include graph caption, X-axis label, and Y-axis label. Figure 24 shows the text entry fields for these legends.

Figure 24. Graph caption and axis caption entry.

After all these steps the user gets one last chance to review all his or her selection. He or she can make changes to them by going back, but cannot change them once he or she hits the Finish button. In Figure 25, one can see all the information collected so far: connection string, data table (actual data source which also may be a view), data columns (continuous columns), categorical columns, and legends.

Then the graph is shown, and the user can interact with the graph. Figure 26 shows the screenshot when the user selects the education value to be "Masters" and the Income-Level to be ">50K". In this example, the census income data set is used [17].

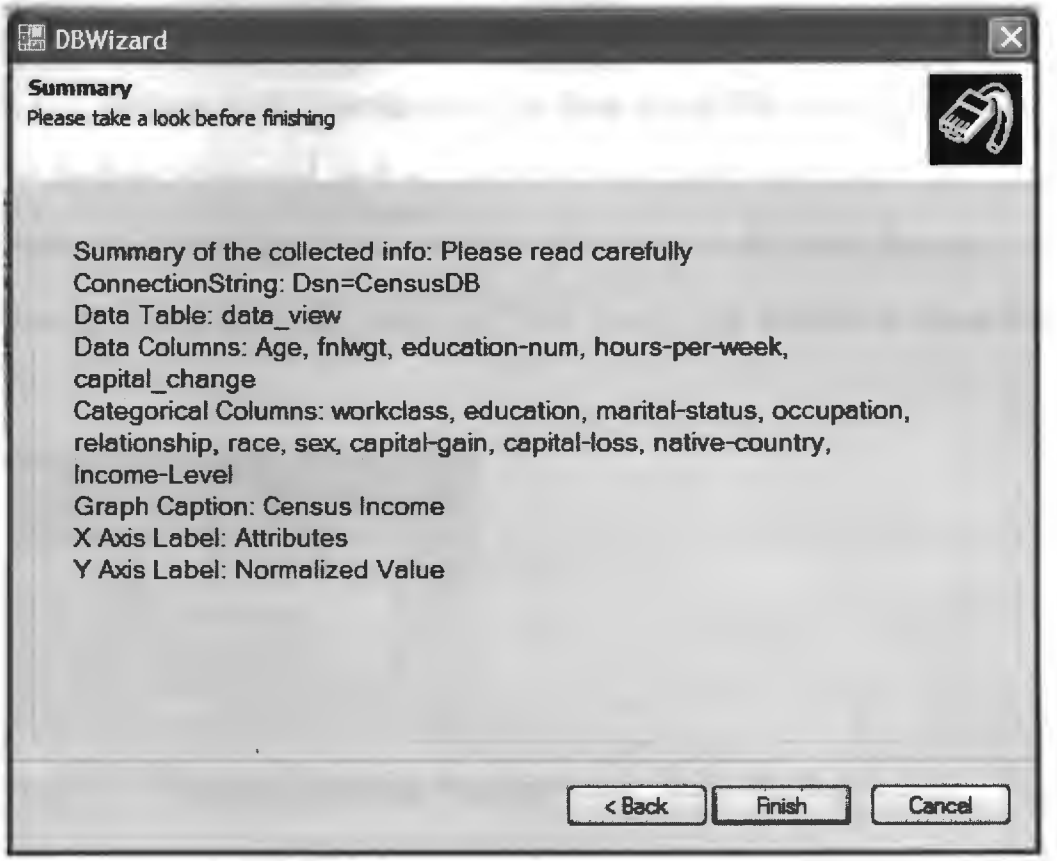

Figure 25. Summary of selected values/options.

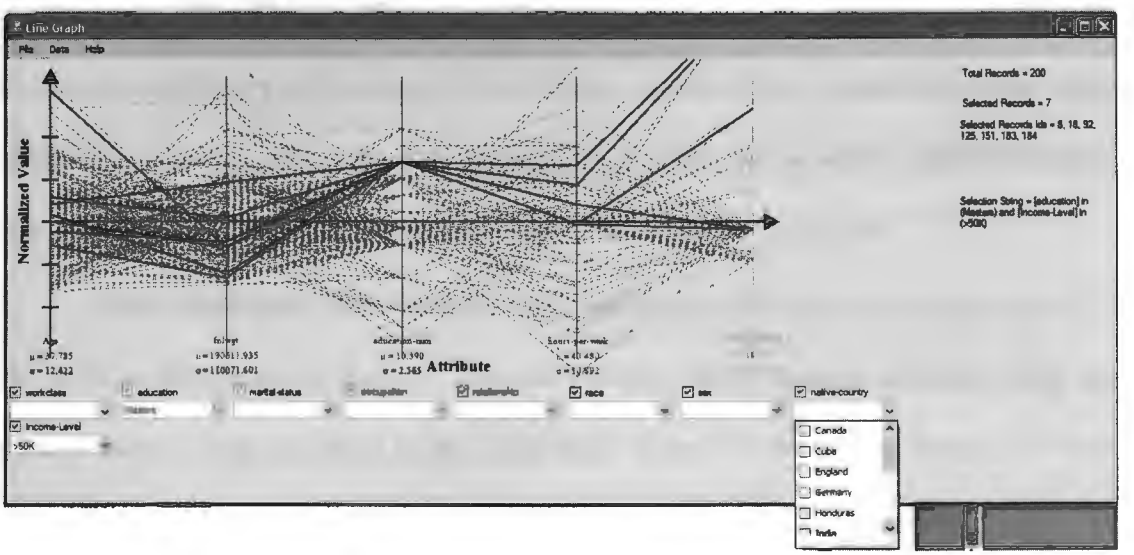

Figure 26. Graph using data from the selected data source (database).

4.2.3. Configuration File Mode

The user loads a configuration file from a local file system which contains all the database information, such as connection string, table name, data fields, categorical fields, graph caption, and axis labels. To work in the configuration file mode, the user goes to "From Config File" from the "Data" menu. This selection is shown in Figure 27.

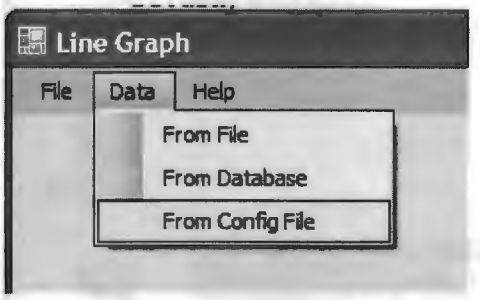

Figure 27. Data menu selecting "Configuration" file as the data source.

Then, the file open common dialog box is shown from which the user selects the configuration file. This configuration file contains the metadata, like graph legends, continuous attributes and categorical attributes to select, the connection string which should be used to connect the data source and the table name which actually contains the data. The configuration file open operation is shown in Figure 28.

After loading the data from the data source specified by the configuration file, the graph is shown on the canvas. The user can now select various attribute values and change them to see the effect in the graph area. Figure 29 shows the screenshot when the user selects the education value to be "Masters" and the Income-Level to be ">50K". In this example, the census income data set is used [17].

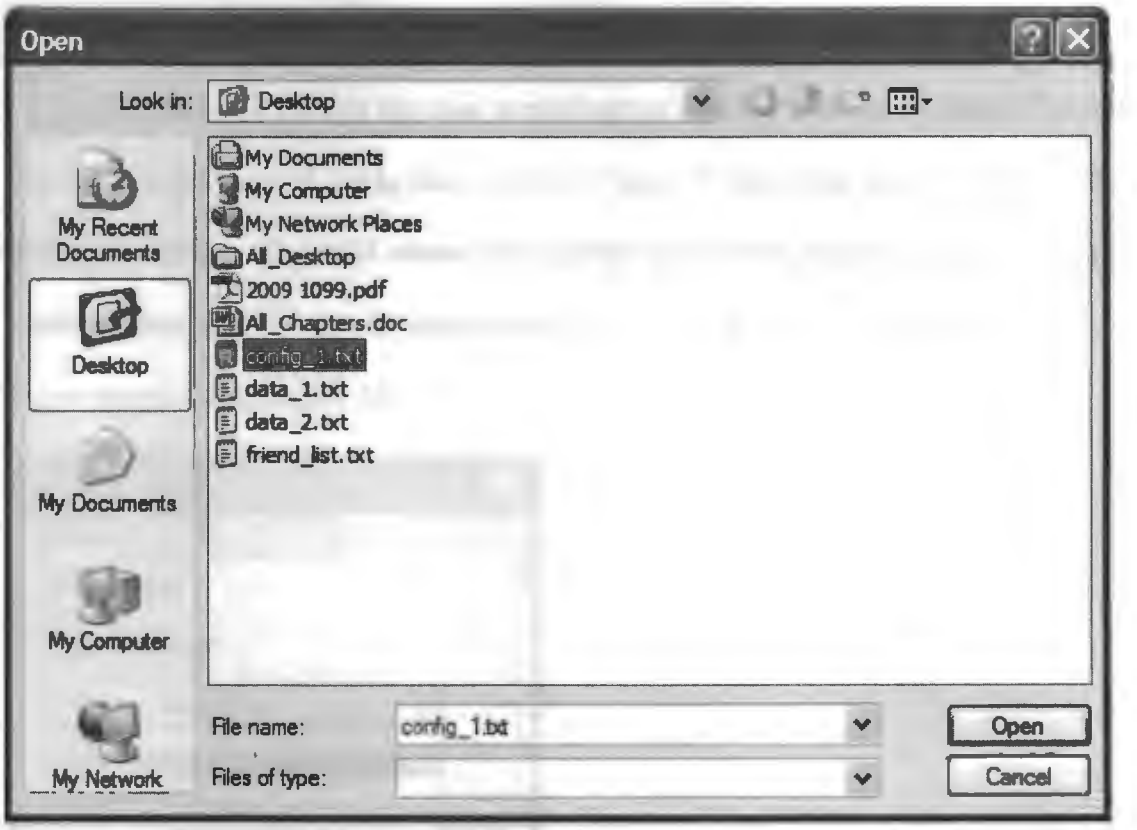

Figure 28. Select configuration file.

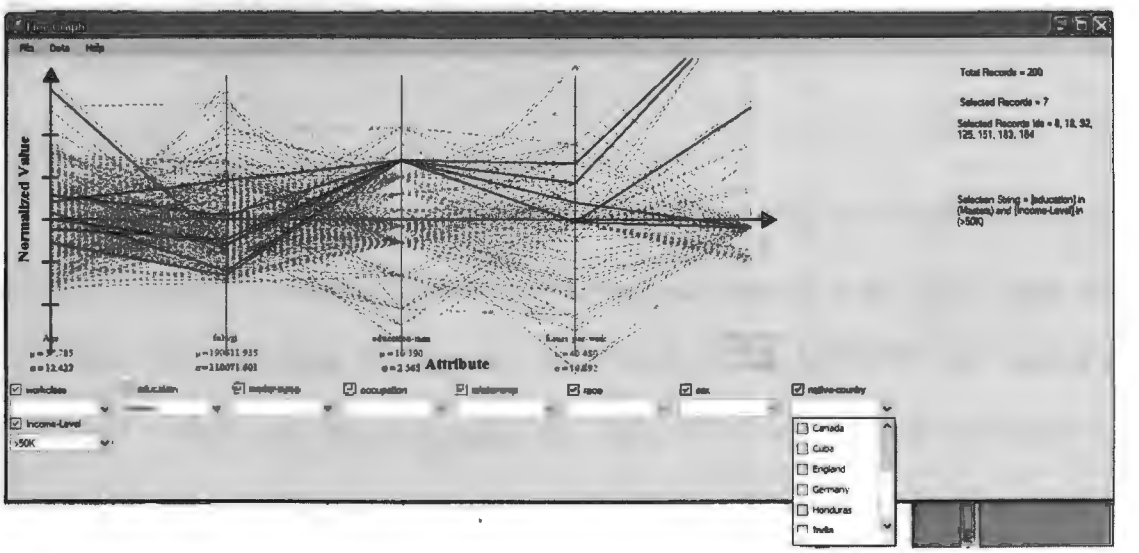

Figure 29. Graph using data from the selected data source (configuration file).

4.2.4. Normalization Type Change

The user can change the data normalization type by selecting "Options" of the "Settings" menu item. A dialog box is shown. Figure 30 shows the available choices for data normalization. Figure 31 shows the resultant graph using range normalization data whereas same graph using Z_normalized (which is statistically normalized) data is shown previously in Figure 16.

Figure 30. Available data-normalization options.

4.3. Web Interface

The web application can be used in a more versatile way than the desktop version. A sample Java applet is developed; it can be embedded in any HTML page and can create the graph from the supplied data. Similarly, an ASP.NET user control is developed to show data from a supplied file. Both of the web solutions can easily be extended to support other types of operation, like loading data from a web form, loading data from a loaded list of files, or loading data from a preloaded list of data source (relational DBMS).

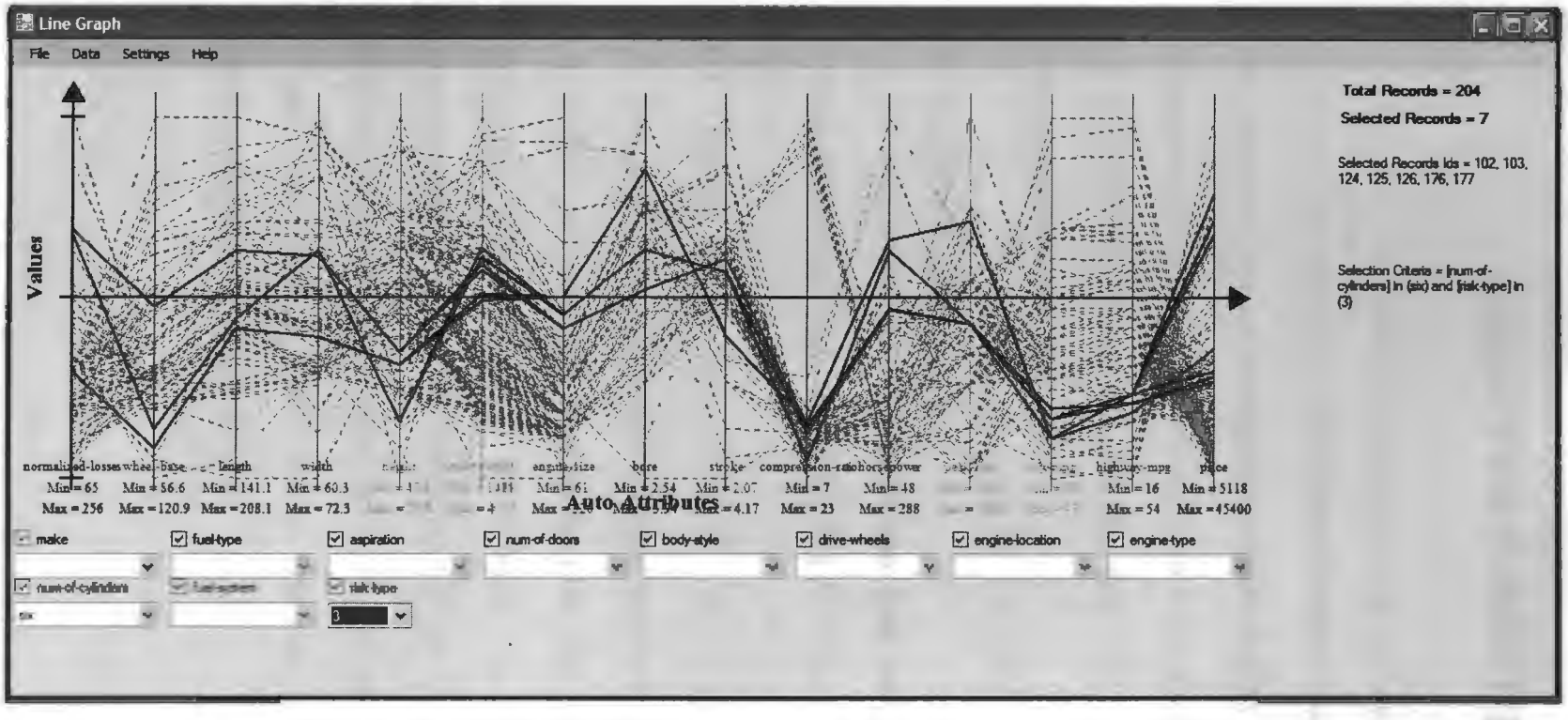

Figure 31. Graph using range normalized data.

CHAPTER 5. RESULTS

This is a very simple, but powerful tool to find the relationships in a subset of data. It gives a user the ability to load data from various data sources and select/unselect various categorical/Boolean attributes to see their effect on the selected subset. Selected data items are shown as thick, blue lines which differentiate them from other data items which are represented as dashed, brown lines. Figure 32 shows a screenshot of the tool depicting various sections of the tool.

5.1. Tool's View Area

The tool's view area is divided into three main sections: i) graph area ii) categorical attribute selection area and iii) data set information area.

5.1.1. Graph Area

The graph area contains the line graphs and attributes as axes. Attributes values are normalized before putting them on the plot. This step helps to compare totally different types of attribute values. The leftmost axis, which is the first continuous attribute, has some markings to identify: the baseline (which is average value μ), $\pm\sigma$, and $\pm 2\sigma$ positions, where σ is standard deviation. Although the graph does not have $\pm 3\sigma$ markings, it approximately considers up to that value to put the point on the graph. Figure 33 shows the graph area separately.

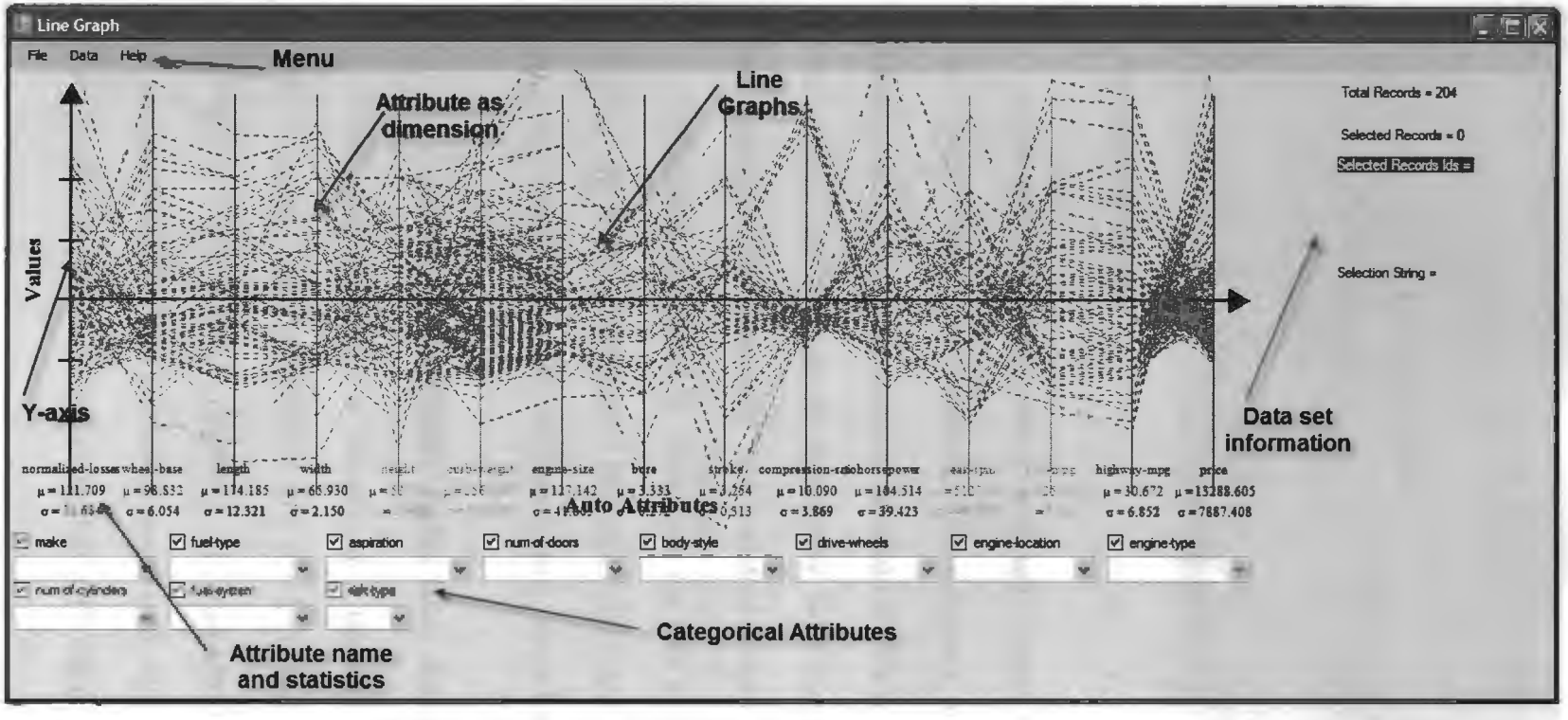

Figure 32. Various sections of the tool.

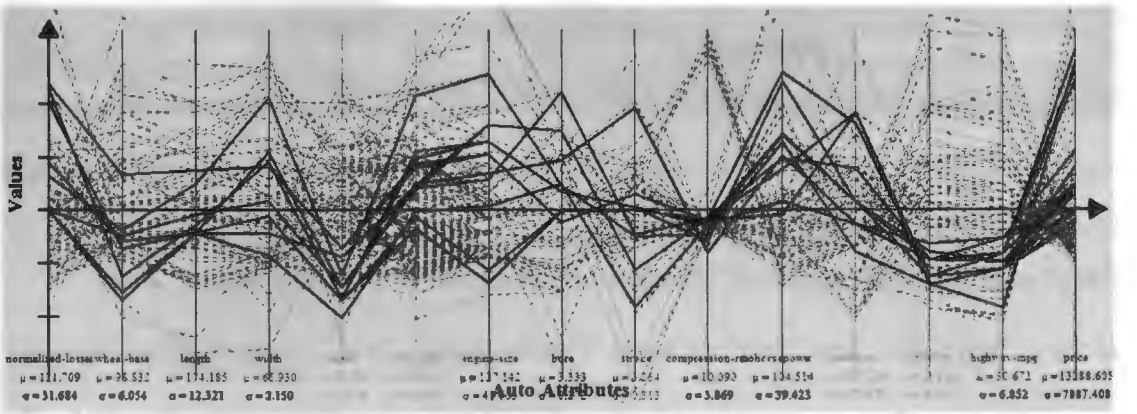

Figure 33. Graph area.

Each attribute axis (dimension) is placed at equal distance and colored so that it can be identified easily. At the bottom of the axis lines, attribute information is placed. Attribute name, average, and standard deviation are the attribute information categories currently shown on this area for statistically normalized data. Other normalizations replace "average" with "maximum" and "standard deviation" with "minimum" values.

Line graphs are drawn by connecting the data points on the axis for each data item. Lines which satisfy the current selection criteria are drawn using thick, blue lines, whereas the rest of the lines are just dashed, thin, brown lines.

5.1.2. Categorical Attribute Selection Area

This section has all the categorical / Boolean attributes. Each categorical attribute is presented as a combination of a checkbox and dropdown combo box, where each combo box item is itself a checkbox. The user can check/uncheck multiple values of an attribute. Also, the user can totally remove the attribute from selection criteria by un-checking the top checkbox. Figure 34 shows the categorical area only.

Figure 34. Categorical attribute area.

Here, "make" category is completely turned off; "fuel-type" is selected to "gas"; "num-of-doors" is set to "two"; "drive-wheels" is set to "rwd"; and "risk-type" is set to "3". Although other attributes are selected at the checkbox, none of their values are selected, so they are not affecting the selection process.

5.1.3. Data Set Information Area

Basic summary and selection information is displayed in this section. It shows the total number of data items, the number of selected data items, the selected data item ids, and selection criteria. Each time the user selects/unselects a categorical value, all the information in this section is modified accordingly. Figure 35 only shows the data set information area.

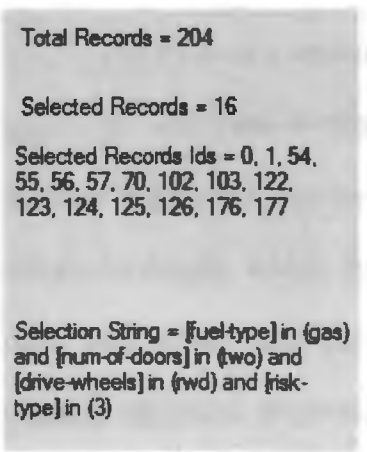

Figure 35. Data set information area.

In Figure 35, the total data items number shown is 204; selected data items is 16; selected record ids are; and a structured query language (SQL) style selection criteria is shown. Both the selected record ids and selection criteria can be copied to the clipboard and may be transferred to another program for reference or interpretation.

5.2. Findings

The test data sets used to develop and validate this tool are the automobile data set and census-income data set from the UCI Machine Learning Repository [17]. Both data sets have some continuous attributes as well as discrete attributes. The automobile data set has 26 attributes; of them, 11 are categorical, and 15 are continuous. On the other hand, the census-income data set has total 14 attributes, with 9 being categorical and 5 continuous data columns. For the German automobile data set, the primary objective is to find some automobile items which may have similar characteristics although they are from different manufacturers.

5.2.1. A Clear Trend

The following selection criteria were found to show very close similarity: risk-type = "3" and num-of-cylinders = "six". This means that cars with six cylinders and with the highest risk type have similar continuous attributes. They have similar property values for height, width, length, carburetor weight, engine size, stroke, compression ratio, horsepower, peak rpm, city and highway mpg, and price. Some properties differ slightly, but most properties are very similar. We can verify these similarities by calculating the correlation coefficient between each data item, i.e., each auto. For this selection, the maximum correlation between selected data items is 0.999505328,

whereas the minimum value is 0.309555563. Figure 36 shows the graph for these selection criteria and also shows the trend in bold lines. Strong correlations are found even though these data items come from different manufacturers. If we exclude Porsche cars and only include Nissan and Toyota, we get a maximum correlation of 0.99876217 and a minimum of 0.815305613. Figure 37 shows the graph for these selection criteria.

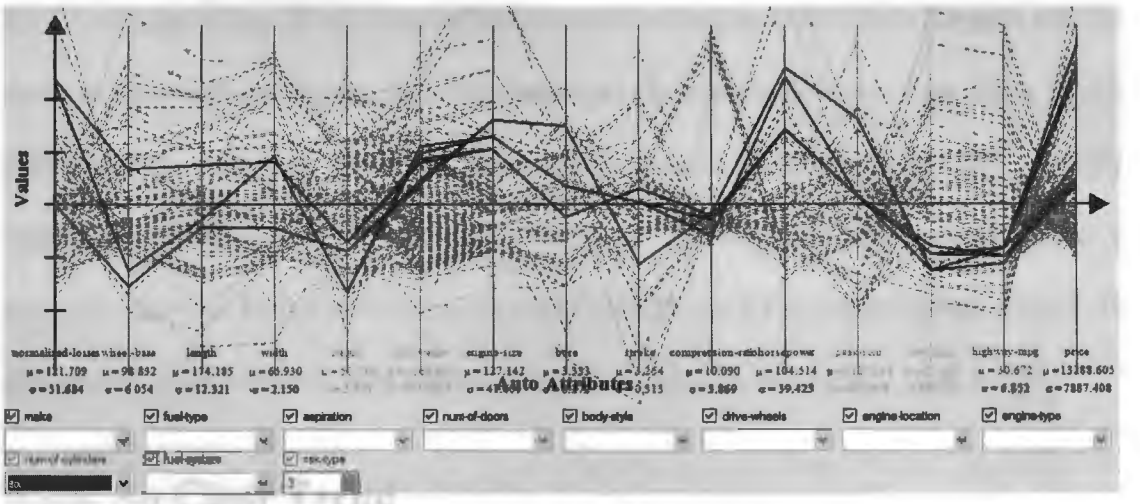

Figure 36. Similar trend in data sub-set (selection criteria: number-of-cylinders="six" and risk-type="3").

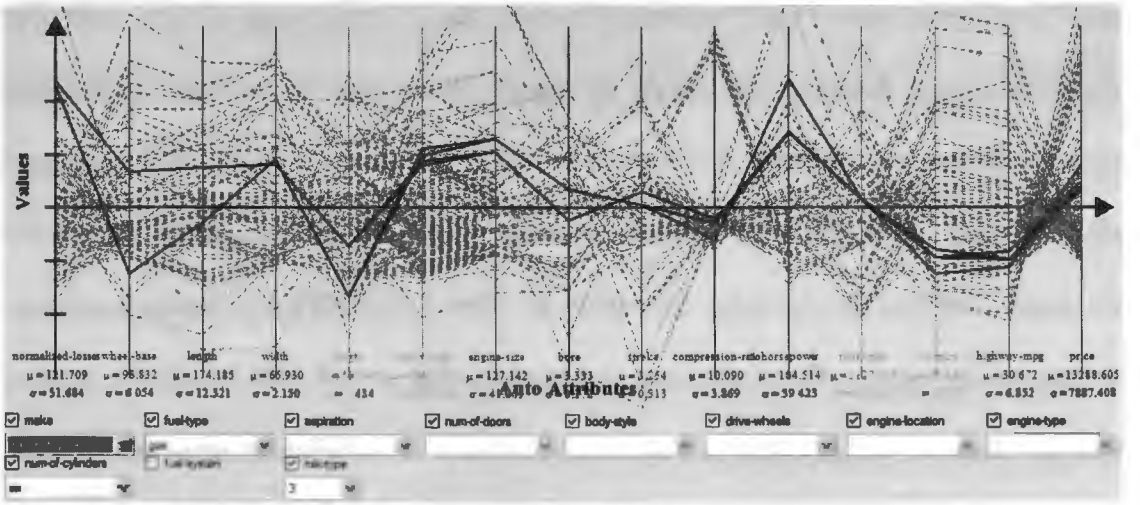

Figure 37. Selection criteria: make ("Nissan" or "Toyota"), number-of-cylinders="six", risk-type="3".

Both Figures 36 and 37 show very similar trends. In Figure 36, the selection criteria satisfy a total of 7 autos from three distinct manufacturers. Most of the axes (dimensions) have similar or close values while some attributes differ significantly. Both Nissan and Toyota have similar normalized-loss property, but Porsche differs a lot. In fact, the set has a missing normalized-loss entry for Porsche which was replaced by the average value. If we remove the normalized-loss property from the data set, the same selection yields Figure 38. The maximum correlation between data items is still 0.999505799, but the minimum correlation improves to 0.427473595, a 38% improvement. Removal of two more attributes (wheel-base and height) yields a dramatic increase in the correlation at 0.585728435, an 89% improvement. Figure 39 shows this graph without three attributes (normalized-loss, wheel-base, and length).

5.2.2. No Clear Trend

Most of the time, no clear trend can be observed. For example, using the selection criteria risk-type="3" and num-of-cylinders="four" does not show many similarities between selected autos. Figure 40 shows the resultant graph. In simple language, four-cylinder autos, which are in highest risk category, do not show much similarity. If the correlation values between selected data items are examined, the maximum value is 0.999900227, and the minimum value is 0.013187844. Here, the minimum value is very low to be considered as a similarity. The larger values are just between the same manufacturers, which is common and expected, but between different makes, the correlation is very low and insignificant.

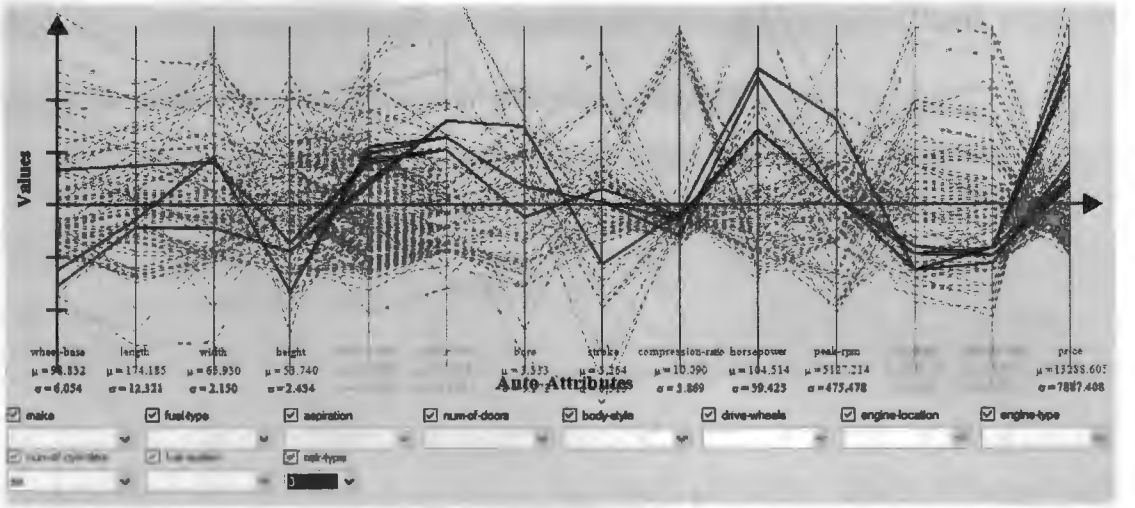

Figure 38. Selection criteria: number-of-cylinders="six" and risk-type="3". (Normalized-loss is removed from the data set.)

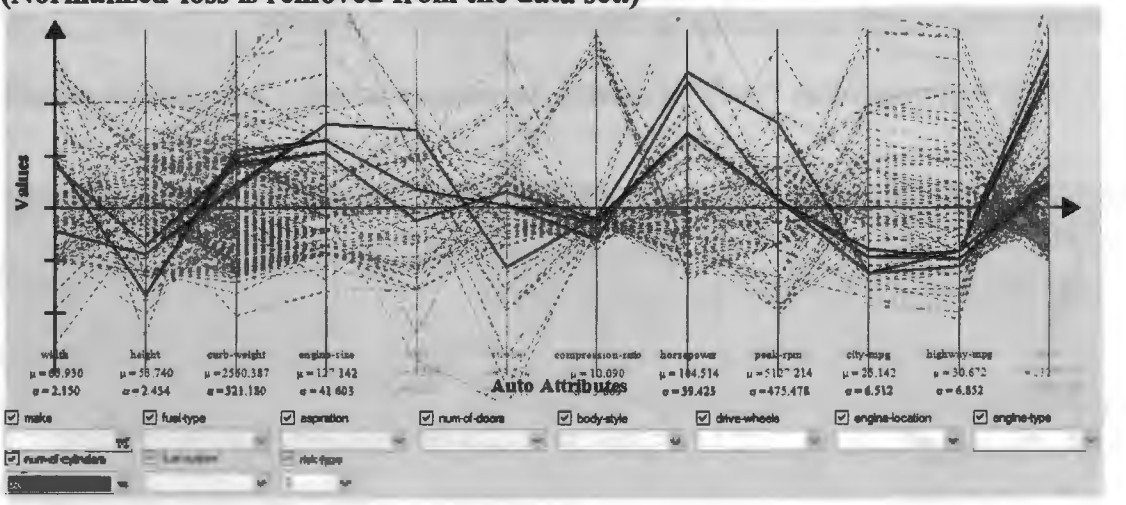

Figure 39. Selection criteria: number-of-cylinders="six" and risk-type="3". (Normalized-loss, wheel-base and length are removed from the data set.)

Sometimes, attributes need to be filtered first. For example, the attribute "fuel-type" partitions the data set into two groups which differ a lot, and it may make less sense to keep them combined. Also, some attributes are directly dependent on others, so one attribute selection may have a huge effect on another. For example, a diesel engine uses a high compression ratio, so selecting fuel-type="diesel" would give a graph where

each selected line passes through the high area of the compression-ratio axis. Figure 41 shows the graph when the "fuel-type" attribute is removed.

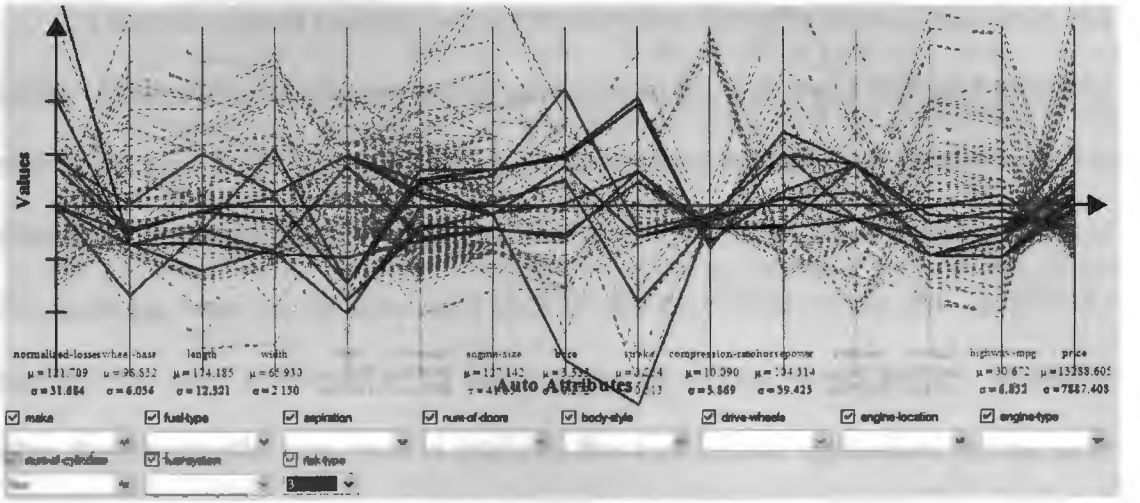

Figure 40. Selection criteria: number-of-cylinders="four" and risk-type="3".

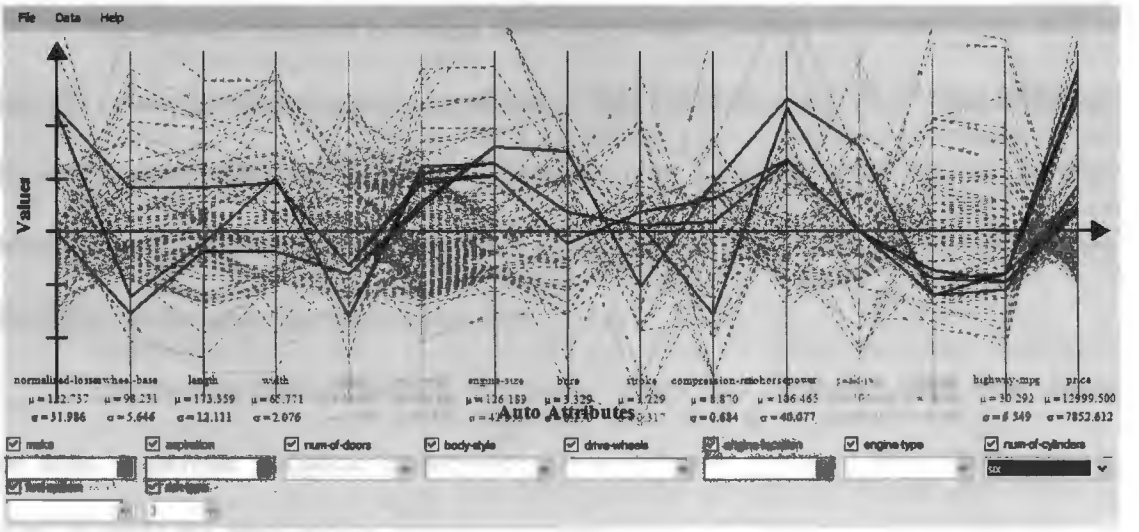

Figure 41. Fuel-type attribute is removed, and all the diesel engine autos are removed (Selection criteria: number-of-cylinders="six" and risk-type="3").

CHAPTER 6. CONCLUSION

I have developed a simple, but powerful tool to show the patterns and trends in a subset of data based on the categorical attributes in the data set. Although I primarily considered gene expression data, this tool can be used to find a quick relationship in other types of data which may also have similar characteristics. For example, figures in this paper are based on the automobile data set and the census-income data set [17]. As this tool has been developed in C# and also ported in Java, it can be very helpful to other developers who may want to integrate this tool in their application. At the same time, this tool can be used as a standalone application from which general users can benefit.

The current solution provides some important features to present data and to identify relationships and trends in a subset of data based on categorical value selection. However, more work can be done to include other facilities which may be helpful. Minimum/maximum customization for a particular attribute, zoom in/out, and scaling facilities can be added for more flexibility.

The current implementation is a purely visual tool. However more work could be done to provide statistical information for a particular group of data which may, in turn, quantify the relationship.

REFERENCES

1. Snezana Savoska, Suzana Loskovska (2009): *Parallel Coordinates as Tool of Exploratory Data Analysis*. 17th Telecommunications Forum TELFOR 2009. Serbia, Belgrade, November 24-26.
2. Scott W Doniger, Nathan Salomonis, Kam D Dahlquist, Karen Vranizan, Steven C Lawlor and Bruce R. Conklin. (2003): *MAPPFinder: Using Gene Ontology and GenMAPP to Create a Global Gene-Expression Profile from Microarray Data*. Genome Biol 4(1):R7.
3. Aravind Subramanian, Pablo Tamayo, Vamsi K. Mootha, Sayan Mukherjeed, Benjamin L. Ebert, Michael A. Gillette, Amanda Paulovich, Scott L. Pomeroy, Todd R. Golub, Eric S. Lander, and Jill P. Mesirov (2005): *Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles*. Proc Natl Acad Sci USA 102(43): 15545-15550.
4. Anne M. Denton, Jianfei Wu, Megan K. Townsend, Preeti Sule, and Birgit M Prüß (2008): *Relating Gene Expression Data on Two-Component Systems to Functional Annotations in Escherichia coli*. BMC Bioinformatics 9:294.
5. Anne M. Denton, Jianfei Wu (2009): Data Mining of Vector–Item Patterns Using Neighborhood Histograms. Knowl Inf Syst 21:173–199.
6. Salim Charaniya, George Karypis, Wei-Shou Hu (2009): *Mining Transcriptome Data for Function–Trait Relationship of Hyper Productivity of Recombinant Antibody*. Biotechnology & Bioengineering 102(6): 1654-1669.

7. Ziv Bar-Joseph, Georg Gerber, David K. Gifford, Tommi S. Jaakkola, and Itamar Simon (2003): *Continuous Representations of Time Series Gene Expression Data*. J Comput Biol 10(3–4):241-256.
8. Darya Chudova, Christopher Hart, Eric Mjolsness, and Padhraic Smyth (2003): *Gene Expression Clustering with Functional Mixture Models*. In: Proceedings of Advances in Neural Information Processing Systems (NIPS).
9. Daxin Jiang, Chun Tang, and Aidong Zhang (2004): *Cluster Analysis for Gene Expression Data: A Survey*. IEEE Trans Knowl Data Eng 16(11):1370-1386.
10. Per Jonsson, Kim Laurio, Zelmina Lubovac, Björn Olsson, and Magnus L. Andersson (2002): *Using Functional Annotation to Improve Clusterings of Gene Expression Patterns*. In: Proceedings of 6th Joint Conference on Information Science, pp 1257-1262.
11. Samuel Kaski, Janne Sinkkonen, and Janne Nikkilä (2001): *Clustering Gene Expression Data by Mutual Information with Gene Function*. In: Proceedings of International Conference on Artificial Neural Networks (ICANN), pp 81–86.
12. Yizong Cheng, George M. Church (2000): *Biclustering of Expression Data*. In: Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology (ISMB), pp 93–103.
13. Giorgio Valentini (2007): Mosclust: A Software Library for Discovering Significant Structures in Bio-Molecular Data. Bioinformatics, 23(3):387-389.
14. Merja Oja, Goran O. Sperber, Jonas Blomberg, and Samuel Kaski (2004): *Grouping and Visualizing Human Endogenous Retroviruses by Bootstrapping Median Self-Organizing Maps*. In Proceedings of IEEE Symposium on

Computational Intelligence in Bioinformatics and Computational Biology. pp 95-101.

15. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten (2009): *The WEKA Data Mining Software: An Update*; SIGKDD Explorations 11(1).
16. Ben Shneiderman, Aleks Aris, Catherine Plaisant, Galit Shmueli, and Wolfgang Jank (2005): *Mosclust: Representing Unevenly-Spaced Time Series Data for Visualization and Interactive Exploration.* INTERACT 2005, pp 835-846.
17. UCI Machine Learning Repository Data Sets [http://archive.ics.uci.edu/ml/datasets.html]: accessed November 2009.
18. Sándor Kromesch, Sándor Juhász (2005): *High Dimensional Data Visualization.* 6th International Symposium of Hungarian Researchers on Computational Intelligence, pp 1-12.

APPENDIX A. INPUT FILE FORMAT AND SAMPLE TEXT FILE

A.1. Text File Format

Format of Text file:

Line1: Graph Caption

Line2: X-axis label

Line3: Y-axis label

Line4: Blank/empty line

Line5: Types of the data columns

Line6: Captions of the data columns

Line7-Line (6+N): Data in each line.

Please note that N is the number of data rows.

A.2. Sample Text File

Census Graph
My Experiments
Log Expression Ratio

number, number, string, string, number, string, string, number
age, education-num, race, sex, hours-per-week, native-country, Income-Level, capital_change

39	13	White	Male	40	United-States	<=50K	2174
50	13	White	Male	13	United-States	<=50K	0
38	9	White	Male	40	United-States	<=50K	0
53	7	Black	Male	40	United-States	<=50K	0
28	13	Black	Female	40	Cuba	<=50K	0
37	14	White	Female	40	United-States	<=50K	0

49	5	Black	Female	16	Jamaica	<=50K	0	
52	9	White	Male	45	United-States	>50K	0	
31	14	White	Female	50	United-States	>50K	14084	
42	13	White	Male	40	United-States	>50K	5178	
37	10	Black	Male	80	United-States	>50K	0	
30	13	Asian-Pac-Islander		Male	40	India	>50K	0
23	13	White	Female	30	United-States	<=50K	0	
32	12	Black	Male	50	United-States	<=50K	0	
34	4	Amer-Indian-Eskimo		Male	45	Mexico	<=50K	0
25	9	White	Male	35	United-States	<=50K	0	
32	9	White	Male	40	United-States	<=50K	0	
38	7	White	Male	50	United-States	<=50K	0	

APPENDIX B. CONNECTION STRING FORMAT AND CONNECTION STRING

B.1. Connection Format

Format of a connection string:

Dsn={DSN Name};uid={User ID};pwd={Password}

Please note that before using this connection string, user needs to create a DSN (Data source name) in his/her machine.

B.2. Sample Connection String

Dsn=MySqlGraph;uid=root;pwd=mypwd

APPENDIX C. CONFIGURATION FILE FORMAT AND SAMPLE CONFIG FILE

C.1. Configuration File Format

Format of Text file:

Line1: Graph Caption

Line2: X-axis label

Line3: Y-axis label

Line4: Blank/empty line

Line5: Types of the data columns

Line6: Captions of the data columns

Line7: Connection string

Line8: Table=<Table name>

C.2. Sample Configuration File

Census Graph
My Experiments
Log Expression Ratio

number, number, string, string, number, string, string, number
age, education-num, race, sex, hours-per-week, native-country, Income-Level, capital_change
Dsn=CensusDB
Table=data_show