Visually Mining Interesting Patterns in Multivariate Datasets

Zhenyu Guo

A PhD Dissertation in Computer Science

Worcester Polytechnic Institute, Worcester, MA

December 2012

Committee Members:
Dr. Matthew O. Ward, Professor, Worcester Polytechnic Institute. Advisor.
Dr. Elke A. Rundensteiner, Professor, Worcester Polytechnic Institute. Co-advisor.
Dr. Carolina Ruiz, Associate Professor, Worcester Polytechnic Institute.
Dr. Georges Grinstein, University of Massachusetts Lowell, External member

Abstract

Data mining for patterns and knowledge discovery in multivariate datasets are very important processes and tasks to help analysts understand the dataset, describe the dataset, and predict unknown data values. However, conventional computer-supported data mining approaches often limit the user from getting involved in the mining process and performing interactions during the pattern discovery. Besides, without the visual representation of the extracted knowledge, the analysts can have difficulty explaining and understanding the patterns. Therefore, instead of directly applying automatic data mining techniques, it is necessary to develop appropriate techniques and visualization systems that allow users to interactively perform knowledge discovery, visually examine the patterns, adjust the parameters, and discover more interesting patterns based on their requirements.

In the dissertation, I will discuss different proposed visualization systems to assist analysts in mining patterns and discovering knowledge in multivariate datasets, including the design, implementation, and the evaluation. Three types of different patterns are proposed and discussed, including trends, clusters of subgroups, and local patterns. For trend discovery, the parameter space is visualized to allow the user to visually examine the space and find where good linear patterns exist. For cluster discovery, the user is able to interactively set the query range on a target attribute, and retrieve all the sub-regions that satisfy the user's requirements. The sub-regions that satisfy the same query and are near each other are grouped and aggregated to form clusters. For local pattern discovery, the patterns for the local sub-region with a focal point and its neighbors are computationally extracted and visually represented. To discover interesting local neighbors, the extracted local patterns are integrated and visually shown to the analysts. Evaluations of the three visualization systems using formal user studies are also performed and discussed.

Acknowledgements

I would never have been able to finish my dissertation without the guidance of my committee members, help from Xmdv group members, and support from my family.

I would like to express my deepest gratitude to my advisor, Matt, for his excellent guidance, patience, immense knowledge, and for the continuous support of my Ph.D. study and research. His advice and suggestions helped me in all the time of doing research, publishing papers, and writing of this thesis. I would like to thank my co-advisor, Elke, who guided me how to conduct thoughtful research and write excellent papers. I am strongly impressed by her enthusiasm and dedicated research attitude, which always encouraged me to seek and perform exciting research to finish my thesis. I would like to thank Prof. Ruiz for her guidance on my directed research. She provided me many useful suggestions to get this work published and this thesis completed. I would also like to thank my external committee member Prof. Grinstein. He devoted a lot time for my comprehensive examination and for my talks. Many thanks for his valuable contributions to this thesis.

I would like to thank Xmdv group members, Zaixian Xie, Di Yang, Abhishek Mukherji, Kaiyu Zhao, and Xika Lin, for the projects and papers we worked together, for the systems we developed together, and also for their broad help during the last four years.

I would also like to thank my parents. They were always encouraging me with their best wishes. Finally, I would like to thank my wife. She was always there supporting me and stood by me through the good times and bad.

This work was funded by the NSF under grants IIS-0812027.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Knowledge discovery in multivariate databases is "the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" [26]. The patterns are generally sub-regions or subsets of data points that meet user requirements or satisfy user demands. We use the term "nuggets" to represent discoveries and patterns, which could be clusters, trends, outliers, and other types of sub-regions or subsets that are of interest to users.

Data mining is an important step of the knowledge discovery process, which consists of particular mining algorithms to extract and detect hidden patterns in the data. Nowadays, many computational data mining techniques have been proposed, and these techniques become more and more automated, however, user intervention and human understanding are still required to discover novel knowledge. This is true, especially when seeking the answers to some complex analysis questions. In those situations, analysts often integrate their expert knowledge, common sense, intuitions into the data mining process [52]. However, in many cases, conventional automated data mining techniques are often treated as "black-box" systems, which only allow very limited or no user interventions. The limitation of pure automated data mining techniques without visualizations has been discussed in [52].

Moreover, in many cases, the discovered patterns and models only make sense and are explainable when it can be visually represented and examined by the analysts. A visualization system that allows analysts to interactively explore the mined patterns, being aware of the relationships between data space and pattern space, is potentially quite powerful. Seeking a more accurate and meaningful pattern, it is more desirable if the users are able to interactively refine and adjust the patterns, based on the users' task and domain knowledge. However, this goal is difficult to achieve if the mining process and the extracted patterns are not explicit to the analysts. The potential advantages of visual data mining tools compared to classical data mining tools are discussed in [52] and [19].

In recent years, visualization has been widely used in many data mining process. It can be used to help the analysts explore and navigate the complicated data structures, re-

veal hidden patterns, and convey the results of data mining [19] [66]. The aim of visual data exploration and mining is to involve the human in the data mining process. Through this, human analysts can apply their perceptual abilities during the analysis, thus gaining a more comprehensive understanding of the mining process and mining results. As discussed in [46], with visual data exploration, the data can be presented in some visual understanding manner, which allows the user to better understand the data, form hypotheses, draw and verify conclusions, as well as perform interactions with the data directly. Keim [45] also argued that visualization techniques are substantially useful for exploratory data analysis and could potentially be very helpful for inspecting large databases, especially in the case where little prior knowledge about the data can be applied.

Visualization can help analysts use visual perception to reveal hidden patterns. The major benefit of visual data exploration is that the users can be directly incorporated in the data mining process. Furthermore, visual analytics can provide an representative and interactive environment, which combines the human's mental cognitive capabilities and computers' computing abilities. This can improve both the speed and accuracy when identifying hidden data patterns. The goal of visual data mining, as detailed in [10], is to help analysts establish in-depth insight of the data, to discover novel and useful knowledge from the data, and to acquire a better understanding of the data.

Keim [44] elaborated on the methodology of visual data mining. They pointed out that using visual data exploration has benefits for users, as they can often explore data more efficiently and obtain better results. Visual data exploration is particularly useful when mining tasks are hard to be done solely by automatic algorithms. In addition, as described in [46], another advantage of using visual data exploration techniques is that users could be more confident about their discovered patterns. These advantages promote a high demand of combining visual exploration techniques and automatic exploration techniques together. A variety of visual data exploration and visual data mining techniques were discussed in [20].

In this dissertation, I discussed three novel visualization systems that facilitate visually and computationally discovering and extracting patterns in multivariate datasets. The extracted patterns can be visually represented for better understanding. The users should be able to interactively adjust the pattern based on the user's task. Visualization systems that integrate the mining process are proposed: from pattern extraction to pattern representation; from pattern examination to pattern refinement.

I list several requirements and desirable features for a visual mining system:

- **Understandability requirement**: conventional computer-supported data mining approaches tend to extract complex and incomprehensible patterns, such as a polynomial regression line, a neural network, or an arbitrarily-shaped sub-region in high dimensional space. These models can be directly used to solve a classification or a prediction task. However, without an explicit representation and human understanding, the results are hard to explain and analyze, especially when the output conflicts with domain knowledge or common-sense. The advantages and disadvantages of the model are hidden from the user, which may mean the user can only

passively perform the mining process and accept the mining results without too much critique.

- **Visual representation**: An effective visualization technique can strongly assist the analysts in discovering hidden patterns and understanding the data mining results. In most cases, the patterns cannot be directly shown and a particular visualization technique should be designed. For example, in XmdvTool, the hierarchical cluster tree and structure based brush [27] provide a good representation of the clustering results. The designed visual representation should clearly reveal the underlying data structures and convey the extracted patterns using visual components, such as color and line width.

- **Refineable and adjustable:** When the extracted patterns are not explicit and visually examinable by the users, they can only generate a new model via adjusting the parameters. However, in most cases, a direct adjustment on the model structure is desirable, for example, removing a branch of a tree structure or changing the coefficient of a regression line.

- **Connection between pattern and data:** The relationship between the pattern space and data space should be clearly presented to the analysts. For example, given a regression model, the user needs to know how well the data points fit the model and which points are the outliers. When the users interactively examine different sub-parts of the model, the data points that fit or correspond to this sub-part should be highlighted.

- **Solve complex real-world application problems:** For analysts, data mining techniques and data mining results are considered as a toolbox for solving the real-world problems or answering task-related questions. An example would be that given a classification model, e.g., a classification tree for classifying the paper acceptance results, the users try to figure out why a certain paper is classified as rejected and how to change the attribute values to make it classified as accepted. This example shows that a data mining pattern cannot be directly used to answer users' guiding questions, except when human intuition and knowledge are involved in the data mining process and pattern exploration.

1.2 Research Goals

In this section, I introduce three topics as my dissertation research goals. Each topic is one type of pattern in multivariate datasets that assists users to understand multi-dimensional phenomena, build models for datasets, and predict target attribute values and class types.

1.2.1 Linear Trend Patterns

The first challenge is to discover and extract linear patterns from a multivariate dataset. Linear trends are one of the most common patterns and linear regression techniques are widely used to mine these patterns. However, the automatic regression procedure and results pose several problems:

- *Lack of efficiency*: When discovering trends in a large dataset, users are often only concerned with a subset of the data that matches a given pattern, so only these data should be used for the computation procedure rather than the whole dataset. Furthermore, locating a good estimation of the trend as an initial input for the regression analysis could expedite the convergence, especially for high dimensional datasets.

- *Lack of accuracy*: Computational results are often not as accurate as the user expects because users are unable to apply their own domain knowledge and perceptual ability during and after discovering models. User-driven modelling and tuning may be required. For example, an extracted linear trend for a dataset with outliers usually tries to cover all the data points, which means it is not an accurate estimation for inliers.

- *Parameter setting problem*: Most model estimation techniques require users to specify parameters, such as the minimum percentage of data points the model includes, maximum error tolerance and iteration count. These are often particular to a concrete dataset, application, and task, but users often don't know conceptually how to set them.

- *Multiple model problem*: If multiple phenomena coexist in the same dataset, many analytic techniques will extract poor models. This is because the computer-based methods try to extract a single model to fit the whole dataset, while in this case, different models for different subsets of data points should be extracted. For example, if the linear trends for males and females are different and coexist in the dataset, a single linear trend doesn't explain the dataset very well. This problem can be solved based on the user's domain knowledge and visual exploration of the dataset.

As part of my dissertation, I developed a system focusing on these problems found in automatic regression techniques. Specifically, I designed a visual interface to allow users to navigate in the model space to discover multiple coexisting linear trends, extract subsets of data fitting a trend, and adjust the computational result visually. The user will be able to select and tune arbitrary high-dimensional linear patterns in a direct and intuitive manner. I designed a sampled model space measurement map that helps users quickly locate interesting exploration areas. While navigating in the model space, the related views that provide metrics for the current selected trend, along with the status of data space, are dynamically displayed and changed, which gives users an accurate estimation to evaluate how well the subset of data fits the trend. The details of this system and the assessing of the technology are discussed in Chapter 3.

1.2.2 Subgroup Patterns

The second difficulty is to discover interesting subgroups in terms of a target attribute and users' requirements from a multivariate dataset. Subgroup discovery is a method to discover interesting subgroups of individuals from a multivariate dataset. Subgroups can be described by relations between independent variables and a dependent variable. An interestingness measure, such as a statistical significance value, is also specified to indicate whether the subgroups are of certain interest. Subgroup discovery is used for understanding the relations between a target variable and a set of independent variables.

The subgroup discovery process poses several compelling challenges:

- *Dynamically submit queries*: since analysts may not know in advance what kind of interesting features the query results have, they may have to repeatedly re-submit queries and explore the results in multiple passes. This makes the mining process tedious and less efficient.

- *Mining results examination problem*: without visual support, users can only examine the mining results in text or tables. This makes it very hard to understand the relationships among different subgroups and how they are distributed in the feature space. A visual representation of the pattern space showing the distribution and relationships among patterns is preferable.

- *Compact representation for visualization*: the mining results are often reported as a set of unrelated subgroups. This kind of mining result is not compact because for the adjacent subgroups, they should be aggregated and clustered when they are of the same interesting type. One benefit could be that an aggregate representation is more compact, which provides the users a smaller report list for easy examination. Another benefit could be that the compact representation can be more efficiently stored in a file and loaded in computer memory.

- *Relationships between patterns and individuals*: without a visualization of the mining results, users cannot build connections between the patterns and the individuals when they explore the mining results. This means that they can only explore the mining result in the form of each subgroup, while they cannot understand the distribution or the structure of the underlying data points.

Focusing on these challenges, our main goal is to design a visual interface allowing users to interactively submit subgroup mining queries for discovering interesting patterns. I proposed and designed a novel pattern extraction and visualization system, called the Nugget Browser, that takes advantage of both data mining methods and interactive visual exploration. Specifically, our system can accept mining queries dynamically, extract a set of hyper-box shaped regions called *Nuggets* for easy understandability and visualization, and allow users to navigate in multiple views for exploring the query results. While navigating in the spaces, users can specify which level of abstraction they prefer to view. Meanwhile, the linkages between the entities in different levels and the corresponding

data points in the data space are highlighted. Details and evaluation of this novel system are in Chapter 4.

1.2.3 Local Patterns

The third challenge is to discover and extract interesting local patterns via sensitivity analysis. Sensitivity analysis is the study of the variation of the output of a model as the input of the model changes. Analysts can also discover which input parameters are significant for influencing the output variable. Although many visual analytics systems for sensitivity analysis follow this local analysis method, there are few that allow analysts to explore the local pattern in a pointwise manner, i.e., the relationship between a focal point and its neighbors is generally not visually conveyed. This pointwise exploration is helpful when a user wants to understand the relationship between the focal point and its neighbors, such as the distances and directions.

We seek to propose a novel pointwise local pattern visual exploration method that can be used for sensitivity analysis and, as a general exploration method, for studying any local patterns of multidimensional data. The primary contributions of this work include:

- *A pointwise exploration environment*: The users should be able to explore a multivariate dataset from a *pointwise* perspective view. This exploration can assist users in understanding the vicinity of a focal point and reveals the relationships between the focal point and its neighbors.

- *A visualization approach for sensitivity analysis*: Sensitivity analysis is one important local analysis method, thus is well suited for our pointwise exploration. The designed local pattern exploration view indicates the relationships between the focal point and its neighbors, and whether the relationship conforms to the local pattern or not. This helps the user find potentially interesting neighbors around the focal point, and thus acts as a recommendation system.

- *Adjustable sensitivity*: The system should allows users to interactively adjust the sensitivity coefficients, which gives users flexibility to customize their local patterns based on their domain knowledge and goals.

Focusing on these requirements, our main goal is to design a visual interface allowing users to perform pointwise visualization and exploration for visual multivariate analysis. Generally, any local pattern extracted using the neighborhood around a focal point can be explored in a pointwise manner using our system. In particular, we focus on model construction and sensitivity analysis, where each local pattern is extracted based on a regression model and the relationships between the focal point and its neighbors. Using this system, analysts are able to explore the sensitivity information at individual data points. The layout strategy of local patterns can reveal which neighbors are of potential interest. During exploration, analysts can interactively change the local pattern, i.e., the derivative coefficients, to perform sensitivity analysis based on different requirements.

Following the idea of subgroup mining, we employ a statistical method to assign each local pattern an outlier factor, so that users can quickly identify anomalous local patterns that deviate from the global pattern. Users can also compare the local pattern with the global pattern both visually and statistically. We integrated the local pattern into the original attribute space using color mapping and jittering to reveal the distribution of the partial derivatives. I evaluated the effectiveness of our system based on a real-world dataset and performed a formal user study to better evaluate the effectiveness of the whole framework. Details and evaluation are discussed in Chapter in Section 5.

1.3 Organization of this Dissertation

The following chapters of this dissertation are organized as follows: Chapter 2 proposes related work of visual data mining and visual analytics. Chapter 3 presents a parameter space visualization system that allows users to discover linear patterns in multivariate datasets. Chapter 3 describes a visual subgroup mining system, called Nugget Browser, to support users in discovering interesting subgroups with statistical significance in multivariate datasets. Chapter 3 discusses a pointwise local pattern exploration system that assists users in understanding the relationship between the selected focal point and its neighbors, as well as in performing sensitivity analysis. Chapter 6 concludes with a summary and the contributions of this dissertation, as well as potential directions for future research.

Chapter 2

Related Work

In this chapter, I will give an overview of visual data mining and introduce related works.

2.1 Visual Data Mining Problem

Data Mining (DM) is commonly defined as "the extraction of patterns or models from data, usually as part of a more general process of extracting high-level, potentially useful knowledge, from low-level data", known as Knowledge Discovery in Databases (KDD) [25], [26]. Data visualization and visual data exploration become more and more important in the KDD process. Analysts use data mining systems to construct their hypotheses about data sets, which rely heavily on data exploration and data understanding. With interactive navigation of multivariate datasets and query resources, *Visual data mining* tools allow the analysts to quickly examine their hypotheses, especially for answering the "what if" questions.

The term *Visual Data Mining* was introduced over a decade ago. The understanding of this term varies for different research groups. "Visual data mining is to help a user to get a feeling for the data, to detect interesting knowledge, and to gain a deep visual understanding of the data set" [10]. Niggemann[51] viewed visual data mining as visual presentation of the data, which is similar to how humans process data presentation. In particular, to understand the data information, humans typically construct a mental model which captures only a gist of the data. A data visualization that is similar to the mental model can reveal hidden information in the data. Ankerst [2] mentioned that visualization works as a visual representation of the data. and moreover emphasized the relation between visualization and the data mining and knowledge discovery (KDD) process. He defined visual data mining as "a step in the KDD process that utilizes visualization as a communication channel between the computer and the user to produce novel and interpretable patterns." Ankerst [2] discussed three different approaches to visual data mining. Two of them involve the visualization of intermediate or final mining results, while the third one, rather than directly being used for showing the results of the algorithm, involves interactive manipulation of the visual representation of the data.

The above definitions consider that visual data mining is strongly related to the human visual understanding and human cognition. They respectively highlight the importance of the three aspects of visual data mining: (a) data mining tasks; (b) visualization for representation; and (c) data mining process. Overall, integrating the visualization into data mining techniques helps convey mining results in a more understandable manner, deepen the end users' understanding about how mining techniques work, and manipulate the mining results with human knowledge.

2.2 Visual Data Mining Process

A visual data mining process proposed in [58] is illustrated in Fig. 2.1. The analyst interacts with each step of the pipeline, shown as the bi-directional arrows that connect the analyst and different mining steps. These links indicate that the human analyst plays an important role in the mining process and can be involved in each step. Indicated by thicker bi-directional arrows, data mining algorithms can also be applied to the data in some steps: (a) before any visualization has been carried out, and (b) after interacting with the visualization.

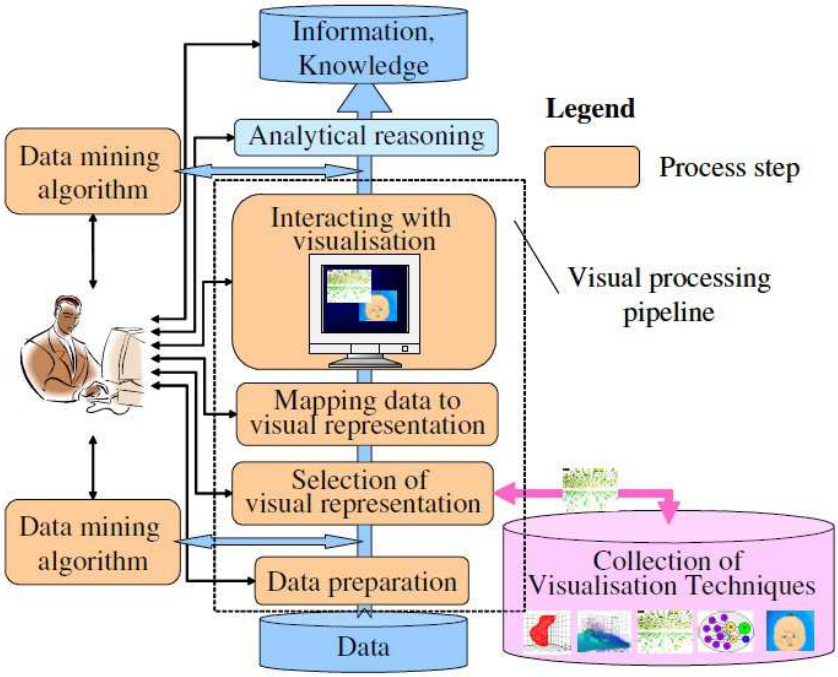

Figure 2.1: Visual data mining as a human-centerd interactive analytical and discovery process [58].

As discussed before, the visual data mining process relies heavily on visualization and interactions. The success of the process depends on the broadness of the collection of visualization techniques. In Fig. 2.1 the "Collection of Visualization Techniques" are com-

posed of graphical representations, each of which has some user interaction techniques used for operating with the representation . For instance, in [17], two visual representations were successfully applied to fraud detection in telecommunication data. Keim [43] emphasized further the importance of interactivity of the visual representation, as well as its link to information visualization.

2.3 Visual Data Exploration for Mining

Many application domains have shown examples where parallel coordinates and scatterplots can be used for exploring the multivariate data. For larger datasets, some user interactions are also incorporated in these techniques, such as selecting and filtering. Inselberg [38] discussed that parallel coordinates transforms the search for relations among different attributes into a 2-D pattern recognition problem. It is also argued that effective user interactions can also be provided for supporting this knowledge discovery process.

The application of a statistical graphics package called XGobi has been described in [59]. They found that visual data mining techniques can be combined together with computational neural modeling, which is a very effective way to detect structures in the neuroanatomical data. This visual data mining tool is used to verify the main hypothesis that neuromorphology shapes neurophysiology. They also discussed that with the feature of brush tour strategy and linked brushing in scatterplots and dotplots, XGobi have been proven as a very successful tool to reveal the hidden structure in their morphology data. As a result, correlation of electrophysiological behavior and certain morphometric parameters are identified and verified.

Hoffman et al. [36] described a case study of using data exploration techniques to classify DNA sequences. Several visual multivariate visualization and data exploration techniques, such as RadViz, Parallel Coordinates, and Sammon Plots [57], have been used to validate and attempt to discover new methods for distinguishing coding DNA sequences from non-coding DNA sequences. Cvek et al. [18] applied visual analytic techniques for mining yeast functional genomics datasets. They demonstrated the application of both supervised and unsupervised machine learning to microarray data. Additionally, they presented new techniques that can be used to facilitate clustering comparisons using visual and analytical approaches. They showed that Parallel Coordinates, Circle Segments [5], and RadViz can help gain insight into the data. [28] and [54] also discussed how visual analytic tools can be applied to Bioinformatics, which indicated that this domain poses many challenges and more and more researchers resort to visual data mining when tackling these challenges.

Recognition of complex dependencies and correlations between variables is also an important issue in data mining. Berchtold et al. [11] proposed a visualization technique called Independence Diagrams, aiming at reveal dependencies among variables. They first divided each variable into ranges. As a result, for each pair of attributes, the combination of these ranges can form a two-dimensional grid. For each cell of this grid, the number of data items in it are stored. The grids are visualized via scaling each attribute

axis. They mapped the the proportional to the total number of data items within that range to the width; and the density of data items in it is mapped to brightness. The authors stated that, with this visual representation, independence diagrams can provide quantitative measures of the interaction between two variables. In addition, it allows formal reasoning about issues such as statistical significance. The limitation for this technique is that for each time, only pairs of attributes can be displayed and analyzed.

Classification is another basic task for pattern recognition in data analysis. Dy and Brodley [23] introduced a technique called Visual-FSSEM (Visual Feature Subset Selection using Expectation-Maximization Clustering). This method incorporated visualization techniques, clustering, and user inter- action to guide the feature subset search by end users. They chose to display the data and clusterings as 2-D scatterplots projected to the 2-D space using linear discriminant analysis. Visual-FSSEM allowed the users to select any subset of features as a starting point, search forward or backward, and visualize the results of the EM clustering, which enables a deeper understanding of the data. In [39], a geometrically motivated classifier is presented and applied, with both training and testing stages, to 3 real datasets. Their implementation allowed the user to select a subset of the available variables and restrict the rule generation to these variables. They stated that the visual aspects can be used for displaying the result as well as exploring the salient features of the distribution of data brought out by the classifier. They tested their classifier on three classification benchmark datasets, and showed very good results as far as test error rates are concerned.

2.4 Visualization of Mining Models

Visualization can also be used to convey the results of mining tasks, which enhances user understanding and user interpretation.

Association rule mining is an important data mining task, which reveals correlations among data items and attribute values. However, understanding the results is not always simple. This is because the mining results are often quite larger than can be handled by humans. Besides, the extracted rules are not generally self-explanatory. Hofmann et al. [37] proposed a method, called Double Decker plots, to visualize the contingency tables to assist the analysts in understanding the underlying structures of association rules. The authors stated that this gives a deeper understanding on the nature of the correlation between the left-hand side of the rule and the right-hand side. An interactive use of these plots are also discussed, which helps the user to understand the relationship between related association rules, for example, for rule sets with a common right-hand side.

Another similar visual representation of multivariate contingency tables is called Mosaic Plots [33]. A mosaic plot is divided into rectangles. The area of each rectangle is proportional to the the number of data items in a cell, i.e., the proportions of the Y variable in each level of the X variable. The arrangement of the rectangles, and how the cells are splitted are determined by both the construction algorithm, as well as the user requirement. The plots reveal the interaction effects between the two variables.

A commercial DM tool called Mineset was introduced by Brunk et al [14]. In integrated database access, analytical data mining, and data visualization into one system to support exploratory data analysis and visualization of mining results. It provided 3D visualization capabilities for displaying high-dimensional data with geographical and hierarchical information. This tool can help identify potentially interesting models of the data using analytical mining algorithms.

Another important data mining results are classifiers that can be used for classification tasks. Some visualization techniques are proposed to support the user's understanding on the classifiers and manipulate the results. For example, Becker et al. [9] discussed a system called Evidence Visualizer to display the structure of Simple Bayes Models, a decision tree model classifier. This system allowed users to perform interactions, examine specific tree node values, display probabilities of selected items, and ask what if questions during exploration. The reasons for the choices of different visualization techniques, such as pies and bars, are also discussed in detail. Kohavi et al. [47] described a visualization mechanism that are implemented in MineSet to display the decision table classifier. Some interactions were provided for exploration of the classifier, such as clicking to show the next pair of attributes, providing drill-downs to the area of interest.

Han and Cercone [32] emphasized human-machine interaction and visualization during the entire KDD process. They pointed out that with the human participation in the discovery process, the user can easily provide the system with heuristics and domain knowledge, as well as specify parameters required by the algorithms. They described an interactive system, called CViz, aiming at visualizing the process of classification rule induction. The CViz system uses parallel coordinates technique to visualize the original data and the discretized data. The discovered rules are also visualized as rule polygons (strips) on the parallel coordinates system. The rule accuracy and rule quality were coded by coloring to render the rule polygons. User interaction was supported to allow focusing on subsets of interesting rules. For example, CViz allows user to specify a class label to view all rules that have this class label as the decision value. The users can also use three sliders to hide uninteresting rules: two to set the rule accuracy threshold and one to set quality threshold.

The Self-Organizing Map (SOM) [61] is a neural network algorithm that is based on unsupervised learning. The goal of SOM is to transform an arbitrary dimensional pattern into a one or two dimensional discrete map, which reveals some underlying structure of the data. SOM involves some adaptive learning process, by which the outputs become self-organised in a topologically ordered fashion. In [62], it is discussed that SOM is a widely used algorithm, and it has led to many applications in diverse domains. The authors also argued that SOM can be integrated with different visualization techniques to enhance users' interpretation.

2.5 Integrating Visualizations into Analytical Processes

Wong [90] argued: "rather than using visual data exploration and analytical mining algorithms as separate tools, a stronger DM strategy would be to tightly couple the visualizations and analytical processes into one DM tool". Many mining techniques incorporate a variety of mathematical steps, where user intervention is required. However, some mining techniques are fairly complex, and visualization plays an important role to support the decision making in the interventions. Standing on this point, the role of a Visual Data Mining technique is considered beyond the traditional belief, that the technique solely participates in some phases of an analytical mining process for exploiting data. Rather, the technique should be viewed as a DM algorithm with visualization as the major role.

A work by Hinneburg et al. is another example that shows the tight coupling of visualization into a mining technique [35]. They proposed an approach to effectively cluster high-dimensional data. The approach was established based on combining OptiGrid, an advanced clustering method, and visualization methods to support an interactive clustering procedure. The approach worked in a recursive manner. Specifically, in each step, if certain conditions are met, the actual data set is partitioned into several subsets. Next, for those subsets which contain at least one cluster, the approach deals with them recursively, where a new partitioning might take place. The approach chooses a number of separators in regions with minimal point density, and then uses those separators to define a multidimensional grid. For a subset, the recursion stops when no good separators can be found. The difficulty in the approach lies in two aspects: choosing the contracting projections and specifying the separators for constructing the multidimensional grid. These two operations have no way to be done fully automatically due to the diverse cluster characteristics in different data sets. The authors resorted to visualization. They developed new techniques that represent the important features of a large number of projections, through which a user can identify the most interesting projections and select the best separators. In this way, the approach improves the effectiveness of the clustering process.

Hellerstein et al. [34] focused on utilizing visualization to improve user control in the process of data discovery. A typical KDD process consists of several steps and requirements, as well as a sequence of user input for submitting queries and adjusting parameters, which are specific to different algorithms. Some examples are the distance threshold for density based clustering, support and confidence for association rule mining, and the percentage of training sets for classification. For these continuous user input, visualization can help ease the process. For example, in a time-consuming task, dynamically setting parameters in real time is a highly desirable ability. Statically setting parameters at the beginning of the process could possibly work less efficiently, as whether the settings are reasonable cannot be known until the end of the process.

Ankerst et al. [4], [3] targeted to the problem that the users are unable to be involved in the middle of a running algorithm. The problem is discussed in a classification task that the users cannot get intermediate results. For most current classification algorithms, users have very limited control to guide and interact with the algorithms. They have no other choices aside from running the algorithm with some pre-set, yet typically hard to

be estimated, parameter values. The users must wait for the final results to tell whether they should have tried some other values. Towards this problem, the authors presented an approach to interactively construct a classifier decision tree. The approach exploits a large amount of visualization for the data set, as well as for the decision tree. Through the enhanced user involvement, the user also gains the benefit of acquiring more insight about the data, during the process of interactive tree construction.

Another similar work is proposed in [65], which emphasized interactive machine learning that involves users in generating the classifier themselves. This allows the users to integrate their background knowledge into the modeling stage and decision tree building process. The authors argued that with the support of a simple two-dimensional visual interface, even common users (not domain experts) can still often construct good classifiers after very little practice. Furthermore, this interactive classifiers construction approach allows users who are familiar with the data to effectively apply their domain knowledge. Some limitations about the approach are also discussed, for example, the manual classifier construction is not likely to be successful for large datasets with large number of attributes to interact with.

Ribarsky et al. [53] propose a mining approach,"discovery visualization". Unlike the other DM tools, the approach emphasizes user interaction and centers on the users. It uses 4D (time dependent) visual display and interaction to a large degree. In order to smooth the user experience, the approach pays a great amount of attention on organizing data, as it facilitates graphical representation, as well as rapid and accurate selection via the visualization. In particular, they present a fast clustering algorithm, that works together with their approach. The algorithm provides users the ability to explore data during continuous adjustment and based on the feedback obtained from the interaction with the visualization. In addition, the algorithm performs fast clustering with the scalability to very large data sets. It also looks beyond direct spatial clustering and completes the task based on the distribution of other variables. As the first step, the algorithm uses an initial binsort to process the data and maintain them into a more manageable size. Initially, the entire (binsorted) data space is viewed as one big cluster. Next, the data set is divided in a iterative manner, until either a user-specified number of clusters have been formed or it makes no sense to perform further division. This approach enables a quick display for a general overview of the data distribution. The user can select regions of interest and perform further exploration.

My research is strongly related to the visual mining ideas, such as exploration for mining and visually knowledge representation. The main goal of my three visual discovery systems is to assist analysts in visually exploring the data space, pattern space, and subgroups to extract and detect certain interesting models or data instances. For example, users are able to explore the parameter space using a linear selection panel to discover strong linear trends, which is discussed in Chapter 3. For each discovered pattern, I design a visual technique, such as a layout strategy of local neighbors discussed in Chapter 5, to help users understand and interpret the extracted knowledge. I also borrow the idea of integrating visualization into mining processes. For example, for the subgroup mining problem mention in Chapter 4, it is difficult to automatically specify the target share range

and subgroup partitioning strategy because of the diverse dataset characteristics. The system allows users to dynamically adjust the cut-point positions for binning, and that target share range for different mining tasks they address.

Chapter 3

Patterns for Linear Trend Discovery

In this chapter, I present a novel visual system that allows analysts to perform the linear model discovery task visually and interactively. This work has been published in VAST 2009 [30].

3.1 Introduction

Discovering and extracting useful insights in a dataset are basic tasks in data analysis. The insights may include clusters, classifications, trends, outliers and so on. Among these, linear trends are one of the most common features of interest. For example, when users attempt to build a model to represent how horsepower x_0 and engine size x_1 influence the retail price y for predicting the price for a given car, a simple estimated linear trend model ($y = k_0x_0 + k_1x_1 + b$) could be helpful and revealing. Many computational approaches for constructing linear models have been developed, such as linear regression [21] and response surface analysis [13]. However, the procedure and results are not always useful for the following reasons:

- *Lack of efficiency*: When discovering trends in a large dataset, users are often only concerned with a subset of the data that matches a given pattern, so only these data should be used for the computation procedure rather than the whole dataset. Furthermore, locating a good estimation of the trend as an initial input for the regression analysis could expedite the convergence, especially for high dimensional datasets.

- *Lack of accuracy*: Computational results are often not as accurate as the user expects because users are unable to apply their own domain knowledge and perceptual ability during and after discovering models. User-driven modeling and tuning may be required.

- *Parameter setting problem*: Most model estimation techniques require users to specify parameters, such as the minimum percentage of data points the model includes, maximum error tolerance and iteration count. These are often particular to

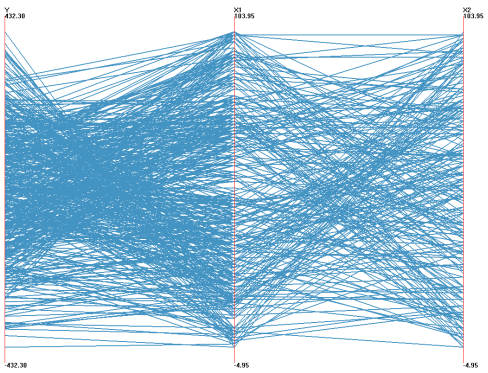

Figure 3.1: A dataset with a simple linear trend: $y = 3x_1 - 4x_2$ is displayed with parallel coordinates. The axes from left to right are y, x_1 and x_2 respectively.

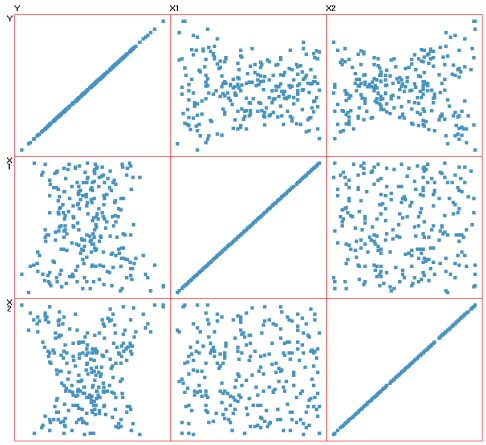

Figure 3.2: A dataset with two linear trends: $y = 3x_1 - 4x_2$ and $y = 4x_2 - 3x_1$ is displayed with a scatterplot matrix.

a concrete dataset, application, and task, but users often don't know conceptually how to set them.

- *Multiple model problem*: If multiple phenomena coexist in the same dataset, many analytic techniques will extract poor models.

Locating patterns in a multivariate dataset via visualization techniques is very challenging. Parallel coordinates [40] is a widely used approach for revealing high-dimensional geometry and analyzing multivariate datasets. However, parallel coordinates often performs poorly when used to discover linear trends. In Figure 3.1, a simple three dimensional linear trend is visualized in parallel coordinates. The trend is hardly visible even though no outliers are involved. Scatterplot matrices, on the other hand, can intuitively reveal linear correlations between two variables. However, if the linear trend involves more than two dimensions, it is very difficult to directly recognize the trend. When two or more models coexist in the data (Figure 3.2), scatterplot matrices tend to fail to differentiate them.

Given a multivariate dataset, one question is how to visualize the model space for users to discern whether there are clear linear trends or not. If there are, is there a single trend or multiple trends? Are the variables strongly linearly correlated or they just spread loosely in a large space between two linear hyperplane boundaries? How can we visually locate the trend efficiently and measure the trend accurately? How can we adjust arbitrarily the computational model estimation result based on user knowledge? Can users identify outliers and exclude them to extract the subset of data that fits the trend with a user indicated tolerance? How can we partition the dataset into different subsets fitting different linear trends?

We seek to develop a system focusing on these questions. Specifically, we have designed a visual interface allowing users to navigate in the model space to discover multiple

coexisting linear trends, extract subsets of data fitting a trend, and adjust the computational result visually. The user is able to select and tune arbitrary high-dimensional linear patterns in a direct and intuitive manner. We provide a sampled model space measurement map that helps users quickly locate interesting exploration areas. While navigating in the model space, the related views that provide metrics for the current selected trend, along with the status of data space, are dynamically displayed and changed, which gives users an accurate estimation to evaluate how well the subset of data fits the trend.

The primary contributions of this research include:

- *A novel linear model space environment*: It supports users in selecting and tuning any linear trend pattern in model space. Linear patterns of interest can be discovered via interactions that tune the pattern hyperplane position and orientation.
- *A novel visualization approach for examining the selected trend*: We project color-coded data points onto a perpendicular hyperplane for users to decide whether this model is a good fit, as well as clearly differentiating outliers. Color conveys the degree to which the data fits the model. A corresponding histogram is also provided, displaying the distribution relative to the trend center.
- *A sampled measurement map to visualize the distribution in model space*: This sampled map helps users narrow down their exploration area in the model space. Multiple hot-spots indicate that multiple linear trends coexist in the datasets. Two modes with unambiguous color-coding scheme help users conveniently conduct their navigation tasks. Two color-space interactions are provided to highlight areas of interest.
- *Linear trend dataset extraction and management*: We present a line graph trend tolerance selection for users to decide the tolerance (maximum distance error tolerance from a point to the regression line) for the current model. Users can refine the model using a computational modeling technique after finding a subset of linearly correlated data points. We also allow the user to extract and save data subsets to facilitate further adjustment and examination of their discovery.

3.2 Introduction and System Components

3.2.1 Linear Trend Nugget Definition

We define a *nugget* as a pattern within a dataset that can be used for reasoning and decision making [67]. A linear trend in *n*-dimensional space can be represented as $(w, X) - b = 0$, where $X_i \in R^n$ denotes a combination of independent variable vector x_i ($x_i \in R^{n-1}$) and a dependent target value y ($y \in R$). Here w and b are respectively a coefficient vector and a constant value ($w \in R^n$, $b \in R$). The data points located on this hyperplane construct the center of the trend. A data point x that fits the trend should satisfy the constraint

$$|(w, x) - b| < \varepsilon$$

Considering that noise could exist in all variables (not just the dependent variable), it may be appropriate to use the Euclidean distance from the regression hyperplane in place of the vertical distance error used above [48]. We define a *linear trend nugget* (LTN) as a subset of the data near the trend center, whose distance from the model hyperplane is less than a certain threshold E:

$$LTN(X) = \{x | \frac{|(w, x) - b|}{\|w\|} < E\}$$

Here E is the maximum distance error, which we call *tolerance*, for a point to be classified as within the trend. If the distance from a data point to the linear trend hyperplane is less than E, it is covered and thus should be included in this nugget. Otherwise it is considered as an outlier or a point that does not fit this trend very well. The two hyperplanes whose offsets from the trend equal E and $-E$ construct the boundaries of this trend. The goal of our approach is to help users conveniently discover a "good" linear model, denoted by a small tolerance and, at the same time, covering a high percentage of the data points.

As the range of the values in the coefficient vector could be very large and even infinite, we transform this linear equation into a normal form to make $\|w\| = 1$ and then represent this vector as S^n, a unit vector in hypersphere coordinates [50] as described in [22]:

$$w_0 = \cos(\theta_1)$$
$$w_1 = \sin(\theta_1)\cos(\theta_2)$$
$$\cdots$$
$$w_{n-2} = \sin(\theta_1)\cdots\sin(\theta_{n-2})\cos(\theta_{n-1})$$
$$w_{n-1} = \sin(\theta_1)\cdots\sin(\theta_{n-2})\sin(\theta_{n-1})$$

Now our multivariate linear expression can be expressed as:

$$y\cos(\theta_1) + x_1\sin(\theta_1)\cos(\theta_2) + \cdots +$$
$$x_{n-2}\sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{n-2})\cos(\theta_{n-1}) +$$
$$x_{n-1}\sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{n-2})\sin(\theta_{n-1}) = r$$

The last angle θ_{n-1} has a range of 2π and the others have a range of π. The range of r, the constant value denoting the distance from the origin to the trend hyperplane, is $(0, \sqrt{n})$ after normalizing all dimensions.

An arbitrary linear trend can now be represented by a single data point $(\theta_1, \theta_2, \cdots, \theta_{n-1}, r)$ in the model parameter space. Users can select and adjust any linear pattern in data space by clicking and tuning a point in the model space.

3.2.2 System Overview

We now briefly introduce the system components and views. The overall interface is depicted in Figures 3.3 and 3.4. The user starts from a data space view displayed with a scatterplot matrix. To explore in the linear model space, the user first indicates the dependent variable and independent variables via clicking several plots in one row. The clicked plots are marked by blue margins; clicking the selected plot again undoes the selection. The selected row is the dependent variable and the columns clicked indicate the independent variables. After the user finishes selecting the dependent and independent variables, he/she clicks the "model space" button to show and navigate in the model space. The points in the data space scatterplot matrix are now colored based on their distance to the currently selected linear trend and dynamically change when the user tunes the trend in the model space. As shown in Figure 3.3, the selected dependent variable is "Dealer Cost" and the two independent variables are "Hp" and "Weight". The points are color-coded based on the currently selected trend; dark red means near the center and lighter red means further from the center, while blue means the points do not fit the trend. Figure 3.4 is the screen shot of the model space view. Each view in the model space is labeled indicating the components, as described in the following sections.

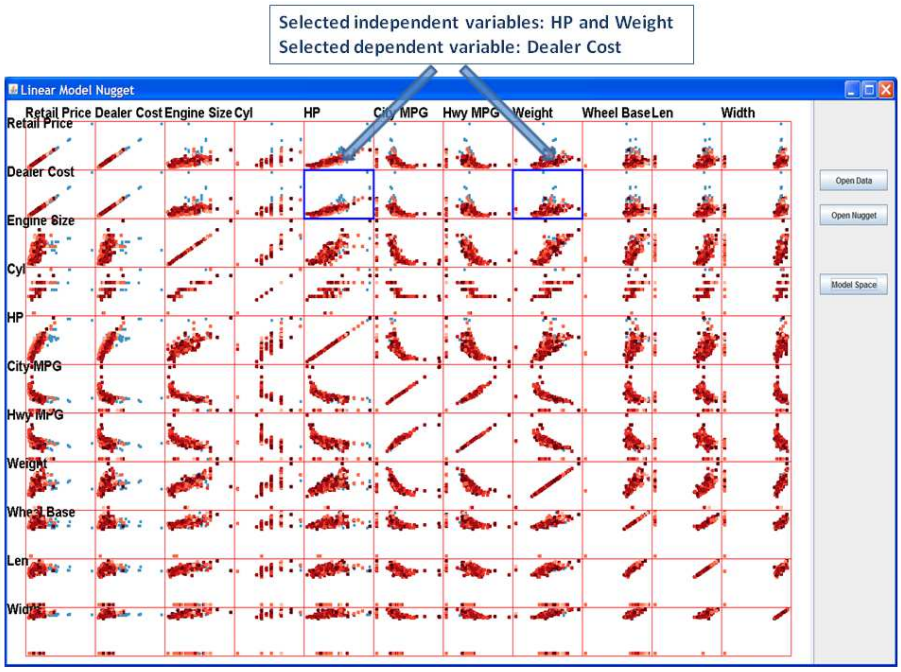

Figure 3.3: The Data Space interface overview.

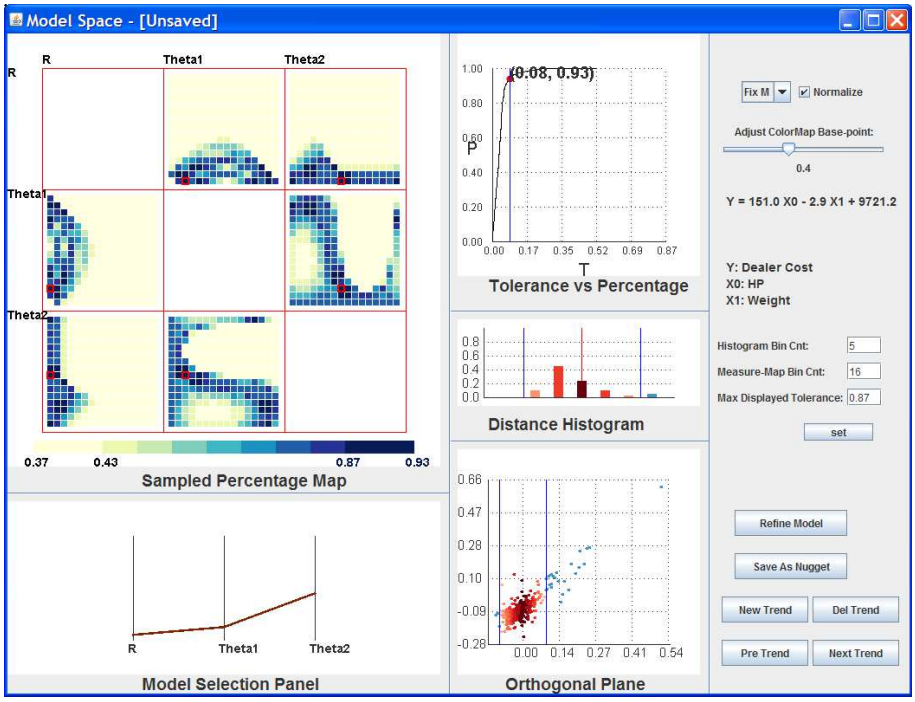

Figure 3.4: The Model Space interface overview.

3.2.3 Linear Trend Selection Panel

We employ Parallel Coordinates (PC), a common visualization method for displaying multivariate datasets [40], for users to select and adjust any linear trend pattern. Each poly-line, representing a single point, describes a linear trend in data space. PC was chosen for its ability to display multiple trends at the same time, along with the metrics for each trend. For example, average residual and outlier percentage are easily mapped to poly-line attributes, such as line color and line width. Users can add new trends, delete trends and select trends via buttons in the model space interaction control panel. Users can drag up and down in each dimension axis to adjust parameter values. During dragging, the poly-line attributes (color and width) dynamically change, providing users easy comprehension of pattern metrics. The parameter value of the current axis is highlighted beside the cursor. This direct selection and exploration allows users to intuitively tune linear patterns in model space, sensing the distance from hyperplane to origin as well as the orientations rotated from the axes. Because the parameters in hypersphere coordinates can be difficult to interpret, the familiar formula in the form of $y = k_0x_0 + k_1x_1 + \cdots + k_{n-1}x_{n-1} + b$ is calculated and displayed in the interface. In Figure 3.5, three linear trends for a 3-D dataset are displayed. The percentage of data each trend covers (with the same model tolerance) is mapped to the line width and the average residual is mapped to color (dark brown means a large value and light yellow means small).

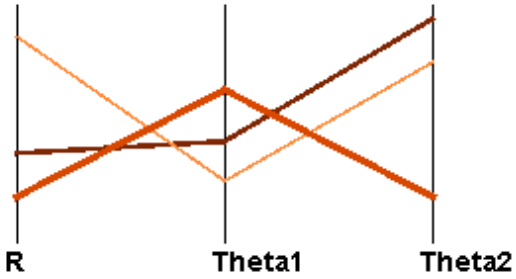

Figure 3.5: The Model Space Pattern Selection Panel.

3.2.4 Views for Linear Trend Measurement

When the user tunes a trend in model space, it is necessary to provide detailed information in data space related to the currently selected trend. Based on this the user can differentiate datasets having linear trends from non-linear trends or without any clear trends, as well as discover a good model during tuning. We provide users three related views for discovering trends and deciding the proper model parameters.

Line Graph: Model Tolerance vs. Percent Coverage

For any multi-dimensional linear trend, there is a positive correlation between the tolerance of the model (the distance between the trend hyperplane and the furthest point considered belonging to the trend) and the percentage of data points this model covers: the larger the model tolerance is, the higher the percentage it covers. There is a trade-off between these two values, because users generally search for models with small tolerance that cover a high percentage of the data. The users expect to find the answer to the following two questions when deciding the model tolerance and percentage it covers: (a) If the model tolerance is decreased, will it lose a large amount of the data? (b) If this trend is expected to cover a greater percentage of the data, will it significantly increase the model tolerance?

To answer these questions, we introduce an interactive line graph for the currently selected model. Model Tolerance vs. Percent Coverage is provided for users to evaluate this model and choose the best model tolerance. It is clear that the line graph curve always goes from $(0, 0)$ to $(1, 1)$, after normalizing. This line graph also indicates whether this model is a good fit or not. If this curve passes the region near the $(0, 1)$ point, there is a strong linear trend existing in the dataset, with a small tolerance and covering a high percentage of the data. This interactive graph also provides a selection function for the model tolerance. The user can drag the point position (marked as a red filled circle in Figure 3.6) along the curve to enlarge or decrease the tolerance to include more or fewer points.

Figure 3.6 shows an example of how to use this view to discover a good model. The

line graph for a linear trend with about 10 percent outliers is shown. The red point on the curve indicates the current status of model tolerance and percentage. From the curve of the line graph, it is easy to confirm that when dragging the point starting from $(0, 0)$ and moving towards $(1, 1)$, the model tolerance increases slowly as the percentage increases, meaning that a strong linear trend exists. After moving across 0.90 percent, the model tolerance increases dramatically while the included point percentage hardly increases, indicating that the enlarged model tolerance is mostly picking up outliers. So for this dataset, the user could claim that a strong trend is discovered covering 90 percent of the data points because the model tolerance is very small (0.07). The corresponding Orthogonal Projection Plane view and Histogram view showing the distribution of data points are displayed in Figure 3.7 and Figure 3.8 (described next).

Projection on the Orthogonal Plane

Given an n-dimensional dataset and an n-dimensional linear trend hyperplane, if the user wants to know whether the dataset fits the plane (the distance from points to the hyperplane is nearly 0), a direct visual approach is to project each data point onto an orthogonal hyperplane and observe whether the result is nearly a straight line.

In particular, we project each high-dimensional data point to a 2-dimensional space and display it in the form of a scatterplot, similar to the Grand Tour [6]. Two projection vectors are required: the first vector v_0 is the normal vector of the trend plane, i.e. the unit vector w described before; the second vector v_1, which is orthogonal to v_0, can be formed similar to v_0, simply by setting $\theta_1 = \theta_1 + \pi/2$. The positions of data points in the scatterplot are generated by the dot products between the data points and the two projection vectors, denoting the distance from the points to the trend hyperplane and another orthogonal plane, respectively. This view presents the position of each point based on their distance to the current trend, which provides users not only a detailed distribution view based on the current trend, but also the capability of discovering the relative positions of ouliers. Figure 3.7 shows the projection plane. The two blue vertical lines denote the two model boundaries. Data points are color-coded based on their distance to the trend center (not displayed). The red points are data points covered by this trend; darker red means near the center and lighter red means further from the center. The blue points are data that are outliers or ones that do not fit this trend very well.

Linear Distribution Histogram

The histogram view displays the distribution of data points based on their distance to the current model. As shown in Figure 3.8, the middle red line represents the trend center; the right half represents the points above the trend hyperplane, and the left half are those below the trend hyperplane. Users can set the number of bins; the data points included in the trend are partitioned into that number of bins based on their distance to the trend center. The two blue lines represent the boundary hyperplanes. The trend covered bars are red and color-coded according to their distance. The color-mapping scheme is the same

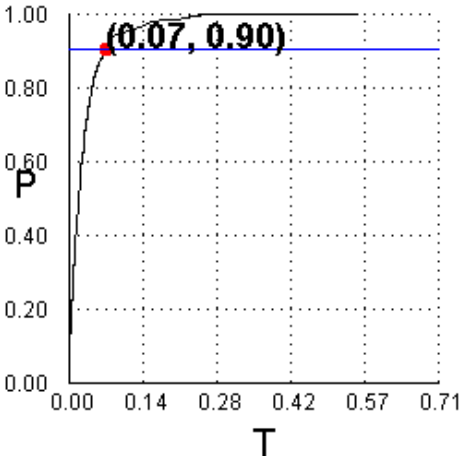

Figure 3.6: The Line Graph of Model Tolerance vs. Percent Coverage.

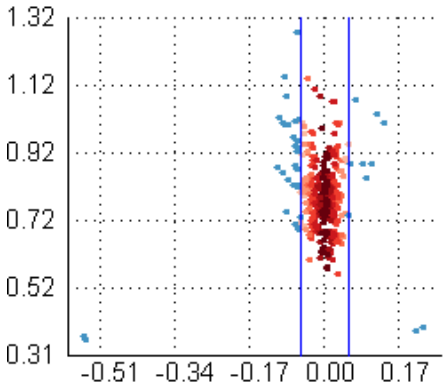

Figure 3.7: The Orthogonal Projection Plane.

as the projection plane view so the user can easily compare these two views. The two blue bars represent the data outside the trend; the right bar is for the data whose position is beyond the upper boundary and the left bar is for the data whose position is beyond the lower boundary.

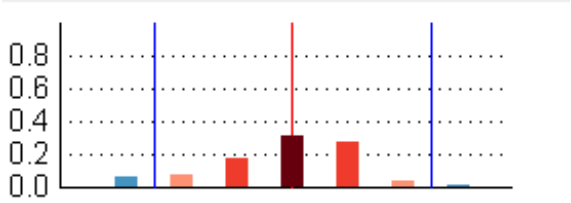

Figure 3.8: The Histogram View.

3.2.5 Nugget Refinement and Management

After finding a good model covering a larger number of data points, the analyst can use a refinement function to tune the model using a computational technique. We employ Least Median Squares [55], a robust regression technique, to compute the regression line based only on the points covered in the current trend, so it is more efficient than basing it on the whole dataset and more accurate because the outliers are not considered. Figure 3.9 shows the user-discovered trend before refinement and Figure 3.10 shows the refinement results.

A linear trend nugget is a subset of data points that lie within trend boundaries. Assuming the user has discovered a trend within several dimensions, it is useful to save it to

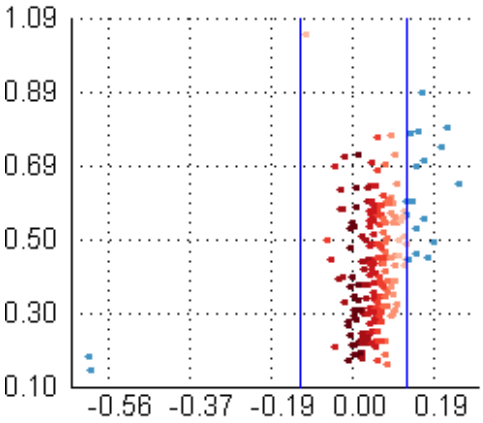

Figure 3.9: The Projection Plane view before refinement.

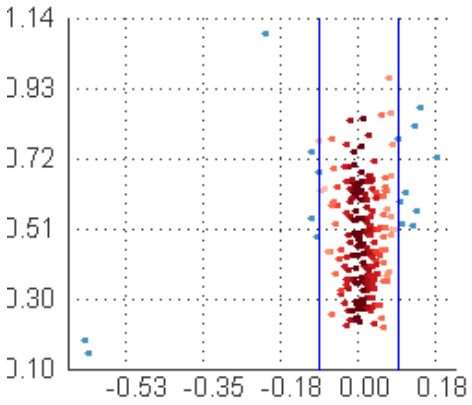

Figure 3.10: The Projection Plane view after refinement.

a file and reload it to examine, adjust and distribute it to other users. After the users find a strong trend, they can extract the points in the trend by saving it as a nugget file. This model selection method is similar to brushing techniques and provides a convenient way for users to identify and exclude outliers that deviate from the trend. This data selection technique is also useful if multiple phenomena are present in the dataset, since the user can save and manage them separately.

3.3 Navigation in Model Space and Linear Trend Model Discovery

3.3.1 Sampled Measurement Map Construction

Even with the metrics of a linear pattern mapped to the poly-line attributes and with the related views for single model measurement mentioned in Section 3.2.2, the user may still feel challenged when searching for good linear trends by tuning the parameter space values, due to the large search area associated with multiple data dimensions. We introduce a sampled model space measurement map for users to view the high dimensional model measurement distribution and navigate in the model space directly and efficiently. The basic idea is that we sample some points in the model space and calculate the measurements for each point (linear pattern), so the user can tune the patterns starting from good parameter sets.

This map is constructed via the following three steps:

(a) We first partition each parameter space variable into several bins. The bin number could be specified by the users. The points in model space located in the center of each combination of bins are selected as sample patterns and the metrics are calculated for

model measuring.

(b) Then we eliminate the patterns with low measurement values and project a high dimensional sampled pattern set to a series of two dimensional pairs. Specifically, for each paired bin position in two dimensions, only the largest measurement (assume larger measurement values denote better models) with the same bin position of these two dimensions is kept as the map value. For example, the bottom left bin in one plot corresponds to the two first bin positions in that dimensional pair, say, bin position 1 for dimension i and bin position 1 for dimension j (the bin number starts from 1). The map value for this position of this dimension pair is selected as the largest measurement in all the sampled patterns whose bin position in the ith dimension and the jth dimension are both 1.

(c) The map values are color-coded based on the metrics. All the pairwise measurement maps are displayed in a matrix view. The initial parameter values are set at the center of the bin with the best measurement, i.e. the minimum tolerance or the maximum percent coverage when fixing the other, which generally provides a good linear pattern for users to start tuning.

The complexity of construction is $PrP_1P_2 \cdots P_{n-1}N$, where N is the size of the dataset;Pr is the number of partitions for r and P_i is the number of partitions for θ_i. The number of partitions for each dimension is defined by users.

Two alternative modes are associated with this view, fixed percent coverage and fixed model tolerance, corresponding to the two measurements for the trends. As mentioned before, the user could change the model tolerance and coverage together in the line graph view. For the first mode, with model tolerance as the measurement, each bin on the map represents a model tolerance with a user-indicated fixed coverage. When the user changes the percentage, this map is dynamically re-calculated and changed (Figure 3.11). For each pairwise bin position in the two dimensional pair, the minimum model tolerance is selected as map value and mapped to color. In this mode, the percentage of points the user wants to include in the trend is designated and users can search for the smallest model tolerances.

The second mode is similar to the first one (Figure 3.12). The difference is we change the measurement to coverage, with a user-indicated fixed model tolerance. This mode is designed for users to specify the maximum model tolerance and search for models that cover a high percentage of points.

For the two modes of measurement map, we use two unambiguous color-coding schemes: (a) model tolerance is mapped from dark red to light pink, with dark red meaning small model tolerance; (b) the coverage is mapped to color from yellow to blue, with blue meaning large coverage.

When the user moves the cursor over each bin, the map value is shown. The bin in which the current model resides is highlighted by a colored boundary. The parameter values are dynamically changed to the bin center, with the largest measurement value as mentioned before, when the user clicks or drags to a certain bin position. This map indicates roughly where good models can be found before tuning the model in the parallel coordinates view. Figure 3.12 shows the coverage distribution map in a 3 dimensional linear trend display. Users can easily find interesting hot spots and drag or click the

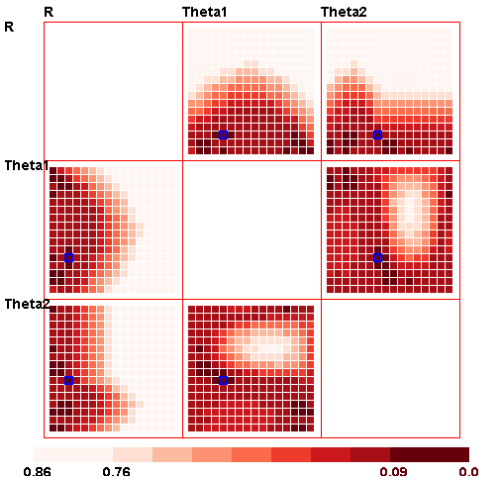

Figure 3.11: The Measurement Map: mode is “fix coverage”.

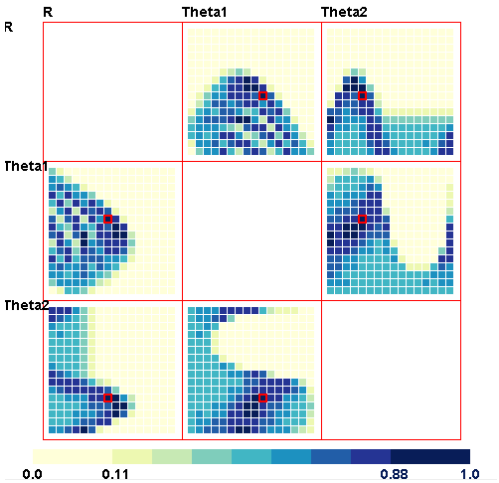

Figure 3.12: The Measurement Map: mode is “fix model tolerance”.

current selected bin into a dark blue area.

3.3.2 Color Space Interactions

It is common that several bins with similar values of interest are shown at the same time in the sampled map near the local maximum, making it hard to locate the best settings. To solve this problem, we provide two interactions in color space.

- Scale the map value to employ the whole color range. Because the values are normalized to $(0, 1)$ and then mapped to color, it is possible that all map values are in a small range; for example, all the coverage values in the map might be located in $(0.7, 1)$ for a very large tolerance in the second mode. In other words, the color map range is not fully used. We allow the user to scale the value range to $(0, 1)$ to use the whole color map.

- Color map base-point adjustment. For the sampled measurement map, the user is only concerned with the high metric values, so a “filter” function to map values less than a threshold to 0 is useful for users to locate the local maximum. In particular, we provide a function for users to change the color map base-point as the threshold. After filtering out the uninteresting areas with low metrics, users can more easily find the positions of good models.

The color space interactions are illustrated from Figures 3.18 to 3.21 and described in Section 3.4.

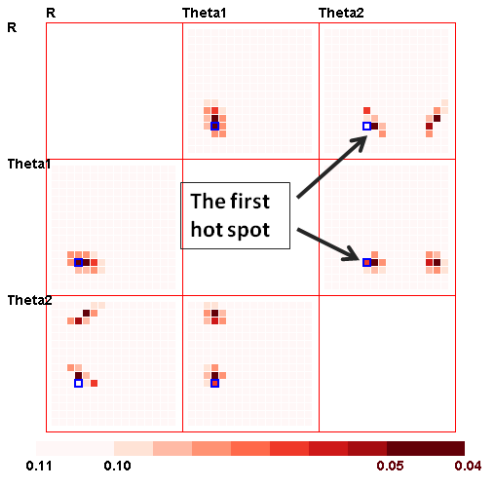

Figure 3.13: The first hot spot is selected representing the first linear trend.

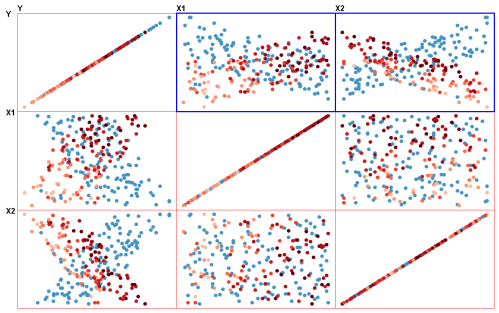

Figure 3.14: The data points that fit the first trend are highlighted in red color.

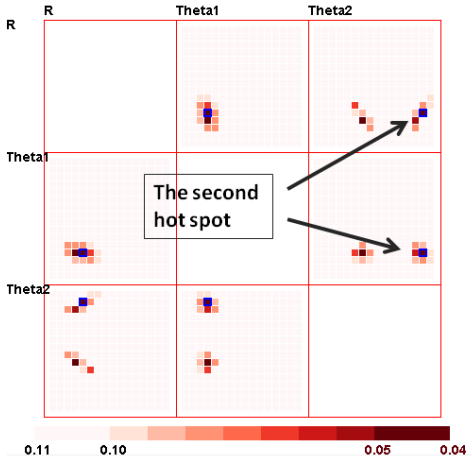

Figure 3.15: The second hot spot is selected representing another linear trend.

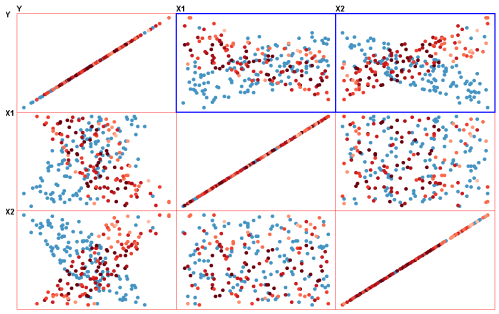

Figure 3.16: The data points that fit the second trend are highlighted in red color.

3.3.3 Multiple Coexisting Trends Discovery

This map is also designed to reveal when multiple linear trends coexist in the dataset, which is very hard to find without visualization. Figure 3.2 shows an example where two linear trends, $y = 3x_1 - 4x_2$ and $y = 3x_2 - 4x_1$ coexist in the three dimension dataset mentioned earlier. Each trend has 50 percent of the data points. When the user fixes the percentage at 0.50, there are clearly two separate hot spot regions indicating two linear trends coexist. Figure 3.13 shows two different hot spots in the sampled map with one of them selected (colored bin). The corresponding subset of data that fit this trend are colored as shown in Figure 3.14. Red means the point fits the model and blue means it doesn't. The other trend and fitting data are shown in Figure 3.15 and 3.16.

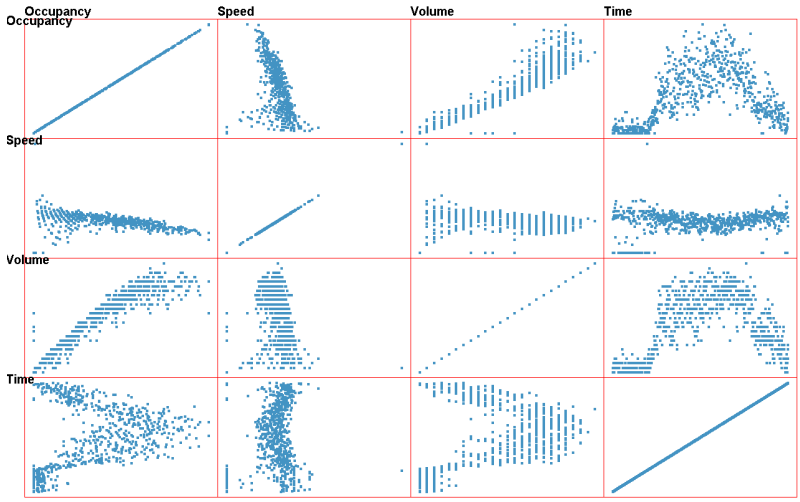

Figure 3.17: Traffic dataset data space view (scatterplot matrix).

3.4 Case Study

In this section, we discuss case studies showing how to discover single or multiple linear trends and construct models for real datasets. The dataset was obtained from the Mn/DOT Traveler Information [49], that collects traffic data on the freeway system throughout the Twin Cities Metro area. Each data point is the traffic information collected by detectors every 30 seconds. The information includes the following variables:

(a) *Volume*: the number of vehicles passing the detector during the 30 second sample period. (b) *Occupancy*: the percentage of time during the 30 second sample period that the detector sensed a vehicle. (c) *Speed*: the average speed of all vehicles passing the detector during the 30 second sample period.

We collected the traffic information for a whole day and added another variable based on the index order to represent the time stamp. Figure 3.17 shows the dataset displayed

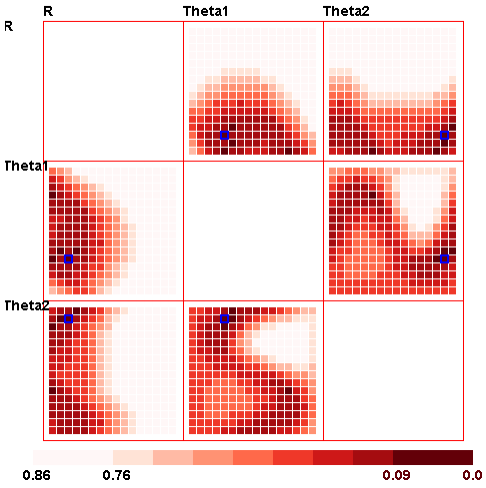

Figure 3.18: The measurement map with the original color range.

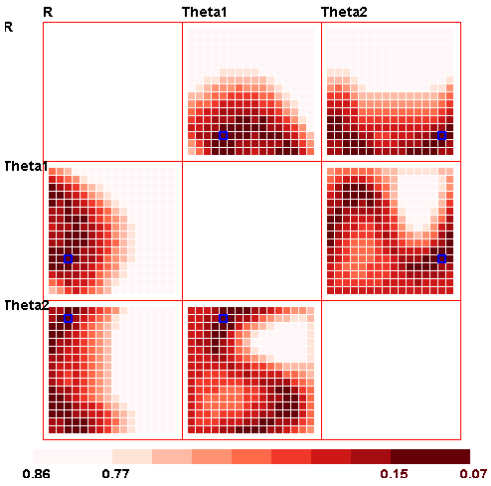

Figure 3.19: After full use of the color map.

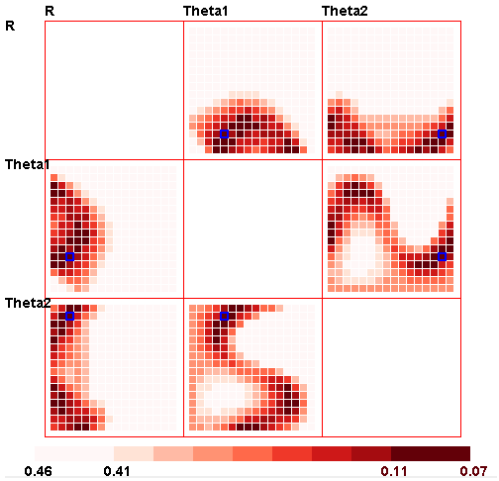

Figure 3.20: Adjust the color map base point to 0.46.

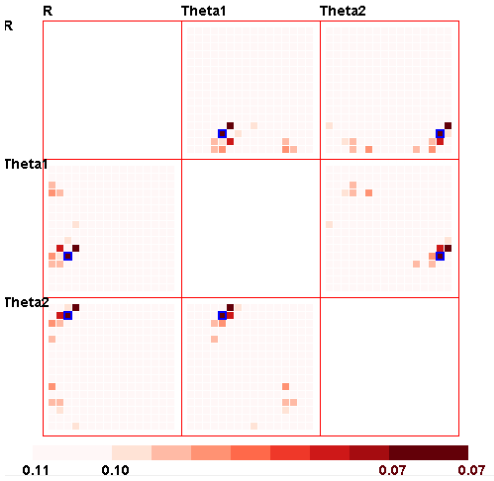

Figure 3.21: Adjust the color map base point to 0.11.

in a scatterplot matrix. Assume a user wants to analyze the correlations between dependent variable occupancy and independent variables speed and volume and construct linear models for these three variables. The aim of this study is to analyze how the average speed and vehicle numbers interactively influence the occupancy. The result is helpful for detecting anomalies, dealing with missing data points and predicting traffic conditions, which can be used for traffic surveillance and control.

If the user wants to build a single linear model to explain the correlations, the first step is to select the view mode and adjust the point on the line graph view to indicate the model tolerance or coverage. Here we use the first mode to discover a model and indicate 85 percent of the data to be covered by the trend, and then search for models with small tolerances.

For further analysis, users can navigate in the sampled measurement map and model selection panel alternately to observe the orthogonal projection plane and histogram to decide whether the current model is a good estimation. To narrow down the search area, the user explores first in the sampled measurement map to click a bin with a good estimation of the model parameters. Notice that the user is only concerned with dark red bins indicating a small tolerance; the user could interact with the color space to fully use the color map and then adjust the color map base-point until the uninteresting areas are eliminated and only red areas remain.

Figures 3.18 to 3.21 show the manipulation details for locating the local maximum value in the sampled measurement map. Figure 3.18 shows the map with the original color range and Figure 3.19 shows the map after fuller use of the color range. Figures 3.20 and 3.21 show the process of adjusting the base point from 0.86 to 0.46 and then 0.11 (shown in the color bar legend). If the map value (tolerance) is larger than this base point, then it will be set to 1 and then mapped to color. From Figure 3.22, the user can easily locate the approximate position of good models and then tune them in the model selection panel.

Figure 3.22 shows the model metric views for the trend in the bin center (model tolerance is 0.07); its corresponding data space view is shown in Figure 3.23. Figure 3.24 shows the adjusted model that fits the data better (model tolerance is 0.05) via tuning the parameter values in the parallel coordinate view; Figure 3.25 displays the data space view.

After refining the model, a good linear estimation for the three variables is constructed: a trend with small tolerance (0.05) covering more than 85 percent of the data points ($y = -0.29x_0 + 1.4x_1 + 25.3$, y: Occupancy, x_0: Speed, x_1: Volume). From the linear equation, we notice that occupancy is negatively correlated with car speed and positively correlated with volume. This three dimensional linear trend plane could also be observed after projection to a two dimensional plane in the data space view displayed by scatterplot matrices. From this we conclude that the more vehicles and the lower speed of the vehicles, the higher percentage of time the detector sensed vehicles, which is fairly intuitive.

Can we use this model to estimate occupancy when we know the speed and vehicle numbers? When we look at the data space view in which the data points are colored according to their distance to the trend, we found this model estimates occupancy well

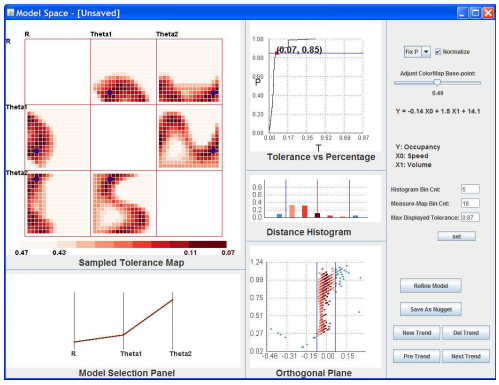

Figure 3.22: The model space view: a discovered linear trend in a bin center.

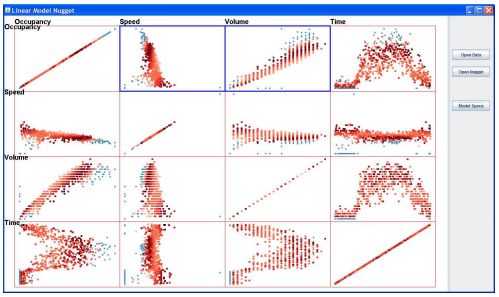

Figure 3.23: The corresponding data space view.

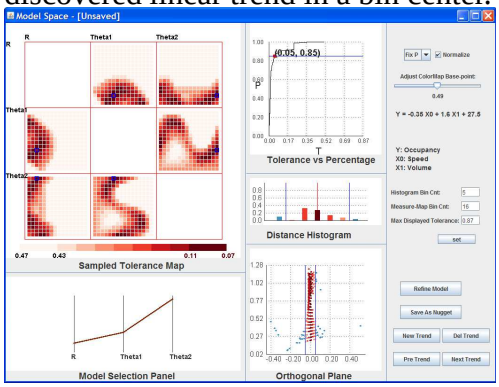

Figure 3.24: The model space view: a better linear trend after user adjustment and computational refinement.

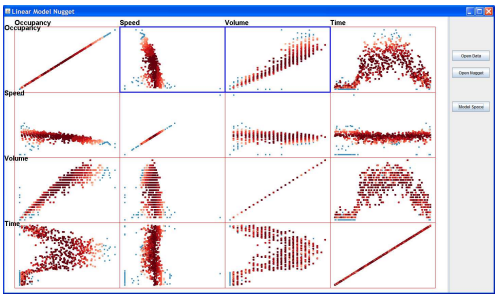

Figure 3.25: The corresponding data space view.

for most of the data points, except the data collected at noon and night. Therefore, a single linear trend could not fit all the data points well, except by increasing the model tolerance to a larger value.

If users want to explain the phenomenon by a single linear trend, the slope of the trend line of occupancy vs. speed does not change for different volume numbers (only the intercept changes). If users want to construct a more complex model with several trends to estimate the occupancy more accurately, multiple linear trends considering different levels of volume can be discovered.

For multiple trend modeling, each trend is not required to cover a large percentage of data points. Conversely, each trend needs to be a strong linear trend represented by a very small tolerance. Therefore, we chose the second mode, i.e. fixed tolerance, and adjust the tolerance to a very small value and then explore in model space as mentioned before. Notice that the value of volume is a discrete number, so it is easy to observe from the Orthogonal Projection Plane view that each subset of data with the same volume value is nearly a straight line in three-dimensional space and the lines are nearly parallel (Fig. 3.32). Thus we adjust the parameter values until each subset of data with a similar volume value aligns to the trend center (Figure 3.32). Adjust the first parameter value (the distance from the hyperplane to the origin) from zero to maximum to extract the data points with different volume values (3 different levels: low volume, median volume and high volume, colored by purple, yellow and red respectively). We can observe from the data space view that different subsets of data reveal different linear trends in the plot of speed vs. occupancy.

We then select two dimensional correlation with occupancy as the dependent variable and speed as the independent variable. We color-code the third dependent variable *volume* with three levels in the orthogonal projection plane view and adjust the parameters to fit different subsets of data with different levels of volume. Figure 3.26 to 3.31 show the three subsets of data fit to different discovered linear trends after refinement in the orthogonal projection plane view and data space view. We can observe from the data space view that as the number of vehicles passing the detector changes, the trend for speed and occupancy alters: the more vehicles passing through, the higher the trend line is and, also the steeper the slope of the trend line. If the volume and speed are known for estimating the occupancy, the user can classify the volume into three bins: low, medium and high, and use different trend lines of speed vs. occupancy to estimate the occupancy value.

How can one explain this model with multiple linear trends for different volumes? If it is ensured that when the detector senses a vehicle, there is only a single car (without any overlapping) passing the detector, then the occupancy is mainly influenced by volume (also influenced a little by speed, but not significantly when volume number changes); it is also clear that low volume indicates low occupancy, which is demonstrated by the lower and less steep trend for speed vs. occupancy when volume is low. But sometimes, especially when volume is large, several vehicles pass the detector together: consider that when two overlapping vehicles pass the detector together, the volume increases but occupancy doesn't. As the volume increases, the occupancy increases, and meanwhile, the

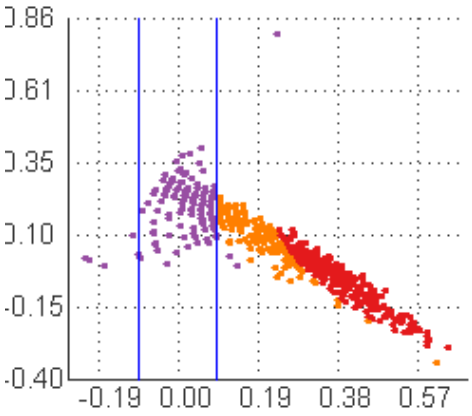

Figure 3.26: Trend fit the data points with low volume.

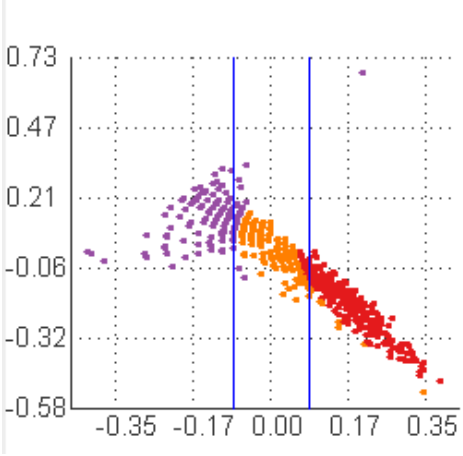

Figure 3.28: Trend fit the data points with medium volume.

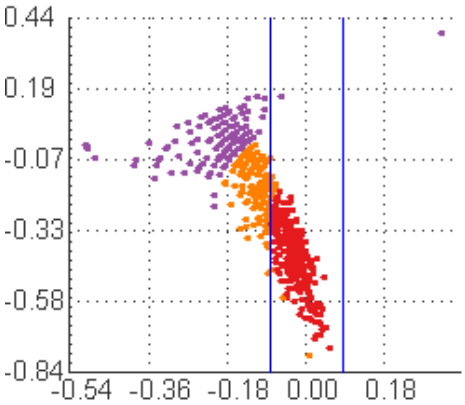

Figure 3.30: Trend fit the data points with high volume.

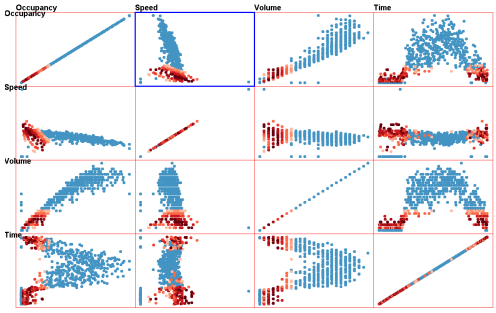

Figure 3.27: Data Space view. The two dimensional trend is $y = -0.11x + 13.8$ (y: Occupancy, x: speed).

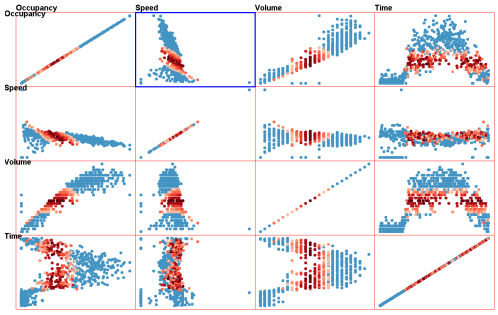

Figure 3.29: Data Space view. The two dimensional trend is $y = -0.17x + 29.7$ (y: Occupancy, x: speed).

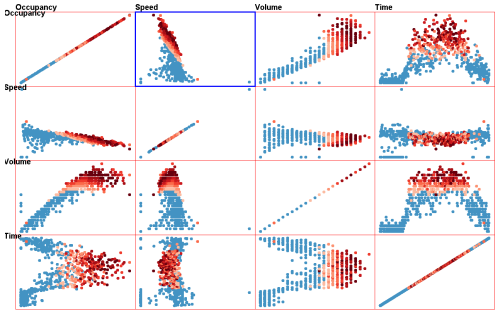

Figure 3.31: Data Space view. The two dimensional trend is $y = -0.38x + 60.2$ (y: Occupancy, x: speed).

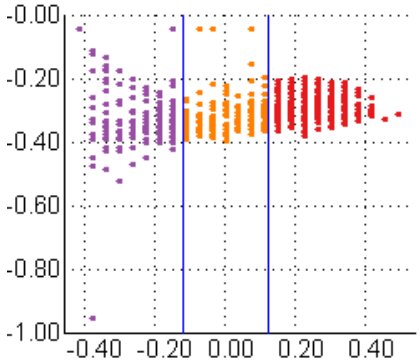

Figure 3.32: The Orthogonal Projection Plane view after adjusting so that data points with similar volume align to the linear trend center. Color coding: purple points are low volume; yellow points are median volume; red points are high volume.

degree of vehicle overlapping increases. When the volume is large, meaning that several vehicles pass the detector together with overlapping, the occupancy is not as predictable just based on volume as it is when volume is small. This suggests the average speed will be more helpful for estimating occupancy. A steeper and higher trend for speed vs. occupancy when volume is large means that occupancy depends more on speed than on volume.

3.5 User Study

In this section, we discuss user studies for evaluating the effectiveness of the visual representations of the parameter space and the user-involved model selection. The user task is to identify linear patterns in multivariate datasets.

The main hypothesis for conducting the user studies is that when outliers exist in a dataset, the linear trend extracted by computational techniques will skew to outliers, which is not a good fit for the inliers, i.e., the data points that are generated from the dominant trend. In this situation, our system can better reveal linear patterns in a multivariate dataset.

To test this hypothesis, two datasets are generated. Each dataset is created using one underlying linear model with a certain randomly added error. Ideally, I would use a real dataset to evaluate the system. However, for real datasets, it is difficult to know the underlying model that generates the population. Therefore, I decided to use synthetic datasets, which is mainly because the underlying model coefficients are pre-defined, thus easier to compare the similarity between the extracted model and the underlying model. For the two datasets, one is easier (dataset A) and the second one is harder (dataset B): the easier one is smaller, has 2 independent variables, and has lower percentage of outliers; while the harder one is larger, has 3 independent variables, and has higher percentage

of outliers. Each user is asked to explore both datasets using our system and report the linear models they discovered. To remove the learning effect, the subjects explore the two datasets with a random order. There are two responses. The first one is the time users spend on exploring in the model space to discover linear patterns. The second one is a linear trend expression found using our system. The linear trends discovered using our system are expected to be more similar to the real underlying model.

There were in total 17 subjects performed this user study. After signing a consent form and given an explanation of the purpose and tasks of this user study, each subject performed the exploration as follows. They first roughly located a good model in the sampled measurement map. The color space interaction can assist them in easily identifying the potential interesting dark regions. After roughly selecting a good trend in the measurement map, they would further refine the model in the model selection panel. During the adjusting of the trend, they could view the projection plane to know what the relationship between the currently selected trend and the data points is, as well as whether the model is a good fit for the data. This refinement consists of two steps: the first one is to adjust the θ values to rotate the trend until the data points lie in a thin vertical position, meaning that the trend is in parallel with the data points. The second step is to adjust the coefficient R value until the data points are in dark red, meaning that the selected linear trend overlaps and fits the data. Notice that the data points mentioned before are only the instances that are generated from the trend, which doesn't include the random outliers. Lastly, the subjects should report the final linear trend expression they found, as well as the time they spent on exploring in the parameter space.

We first compare the means of squared errors (MSE) for these two datasets. For dataset A, the squared error that linear regression was 1.555; while the mean value of MSE for subjects was 1.752. Since a lower MSE value means better, when considering this metric, linear regression produced better results. We found a similar result for dataset B: the error from linear regression was 1.149, and the subjects got 1.22 (mean value). The reason why the subjects got worse results is that linear regression always gets the linear trend that minimizes MSE. However, this minimized MSE also takes the randomly distributed outliers into consideration. In this case, the extracted linear model is distorted by the outliers, which means it is not a good fit for the inliers.

We then compared the user detected model with the real underlying model. Each linear trend can be viewed as a single instance in the model space. Since these synthetic datasets are generated from pre-defined models, we can use the Euclidean distance to the real model as the metric, which is better to measure the goodness of the model. For dataset A, the Euclidean distance for the linear regression is 1.33, while the mean value of this metric that the subjects got is 0.64. The smaller value means the subjects got a more similar trend to the real model. The one sample t-test shows that we can reject the null hypothesis that linear regression is better than the user-driven method (p-value is 0.01). For dataset B, we also got similar results: the linear regression error was 0.785 and the mean value of this metric that the subjects got was 0.68. Although the p-value (0.086) suggests that the difference is not as significant as appeared in dataset A at the significance level of 0.05, the subjects still got a better result, compared to the linear

regression method. From these user study results, we showed that without detecting and removing outliers in the datasets, the linear regression technique extract linear models that are influenced by noise, which is not a good fit for the dominant trend. The proposed model space visualization can better help the users detect outliers and extract a better model. We will discuss later the challenge of using automatic techniques to detect and remove outliers before extracting linear models.

The user study result shows that for the higher attribute space, the subjects did not perform as good as they did in the lower dimensional space. Figure 3.33 shows the comparison of the time the subjects spent on those two datasets. A paired t-test indicates that the subjects spent significantly more time on the harder dataset (the one with more independent variables). The p-value is lower then 0.001. The reason why the subjects spent more time and detected a less accurate result is mainly because for more independent variables, they have a larger model space to explore, as well as there were more parameters to adjust. These results indicated that the system doesn't scale very well as the dimensionality increases. This limitation also appeared during the sampling in the model space: for 6 independent variables and each variable is sampled at 4 positions, the whole sampling process took more the 5 minutes to finish. This is because the number of sampling points is exponential to the number of dimensions. We also infer that for more than 6 coefficients, it would be very difficult for the users to tune on each axis until they find a strong linear trend. Here are some potential improvements to solve this issue. One is to allow the users to specify a sub-region in the parameter space that is interesting to them and re-sample in this sub-region with a higher sampling rate. Another improvement could be to allow the users to interactively and progressively mark and highlight outliers, and later extract linear trends using only inliers with computational techniques.

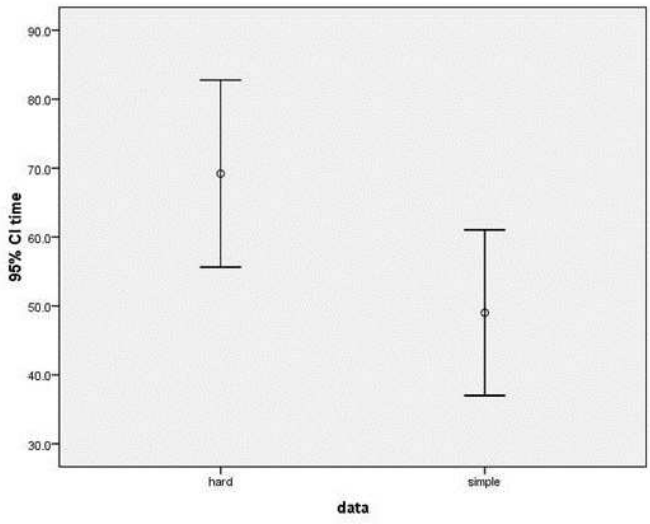

Figure 3.33: The comparison of the time the subjects spent on the two dataset: simple means dataset A and hard means dataset B.

We also examined the relationship between the time the subjects spent and the error

they made. Figure 3.34 shows the scatterplot for these two values. Each point is the response of a single subject. It is clear that except for some outliers, subjects who spent a fairly long time also got more accurate results. For most subjects, there is a negative correlation between these two responses. This means that if the subjects have patience and would like to spend more time to explore in the model space, they tend to get better linear models with lower errors. Based on this result, we may infer that after being trained and using this system for a long time to get familiar with the system, the users can get better linear models with less time, compared to the novice users. This needs to be verified in a future longitudinal user study.

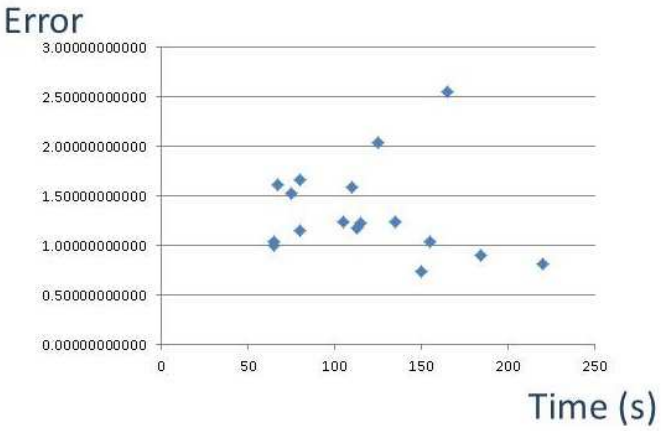

Figure 3.34: The scatterplot for time and error. Each point is one subject. A negative correlation can be seen for these two responses.

Since automatic techniques can also detect outliers for these datasets, a linear regression can be performed after removing the outliers. Using this method, the extracted linear pattern could be better, compared to without removing outliers. However, since the number of outliers is unknown, the users have to go through the procedure: first try to remove a certain number of outliers; then observe the resulting model. After several of such trials, the users can finally identify a reasonable number of outliers, and extract the linear model for the dataset. Also, since most outlier techniques require the users to specify some parameter settings, the users have to try different parameter settings to get a set that meets this requirement. In order to understand how the users perform this parameter setting task, we invited a subject to extract linear patterns after removing outliers in the dataset. He used Weka to detect outliers using the k-NN method. This method requires two parameters: a radius distance and the number of neighbours. Without knowing the percentage of outliers, he had to try several percentages of outliers: from 0 percent to 15 percent. In this study he used dataset B to detect a linear trend. The subject tried more than 20 parameter settings and spent more than 40 minutes to find a set of parameter settings that can result in a reasonable percentage of outliers. For each parameter setting, the subject removed the resulting outliers and created a new data set merely using the inliers. Therefore, a set of new datasets were created, and for each of which, he then extracted lin-

ear pattern using the regression technique and recorded the errors. Since removing more data points generally results in a smaller error rate, he decided to try different percentage values from smaller to larger, until he thought the error didn't decrease dramatically. It costed him over 20 minutes to generate new datasets and extract linear patterns for them. The final model he used is the one resulting from removing 5 percent as outliers. The error using this model is 0.536, which is better than the the mean error using our system. However, as discussed before, the whole process required more than one hour. Compared to our system, on which the subjects spent less than 2 minutes on each dataset, the traditional method appears to be a very time consuming and tedious task. Also, without visual exploration of the dataset, the subject cannot understand the extracted model, i.e., the relationship between the linear trend and the dataset, such as how well the trend fits the data and what is the appropriate tolerance to reject the points as outliers.

Another user study we performed is to evaluate whether our system can assist analysts in discovering linear patterns when multiple linear trends coexist in one multivariate dataset. When multiple linear patterns exist in the dataset, using one linear trend to fit the data tends to give a poor result. However, if the analysts can notice this and use multiple trends to fit the data, better results can be achieved. Since the number of trends is the key thing to decide, the major challenge here is to identify the number of trends in the dataset. We wanted to examine whether analysts can identify the correct number of trends when a small number of trends exist. If they can tell this, they can further use our system to separate different trends, or use other computational techniques to extract multiple trends. To test this, 6 datasets were generated. Each dataset was generated using zero, one, or two linear models. The 6 datasets were randomly partitioned into two groups. Each subject explored one group of datasets and report how many linear trends they discovered for each dataset. Which group was assigned to the subjects was also randomized. The result shows that the accuracy is more the 0.9. This means that using our system to visually explore the data can largely assist the analysts in understanding datasets and accurately estimate the number of linear phenomena. The average time spent in exploring each dataset by the subjects was 29 seconds.

3.6 Conclusion

In this chapter, we described a novel model space visualization technique to support users in discovering linear trends among multiple variables. Using this system, analysts can discover linear patterns and extract subsets of the data that fit the trend well by navigating in the model space and building connections between model space and data space visually. The case studies show how our system can be used effectively to reveal single and multiple linear trends and to build explanation models for multivariate datasets. We performed user studies to evaluate the effectiveness of the user-driven linear trend discovery when outliers and multiple trend exists in the datasets. We also compared the visual and computational methods for extracting linear patterns when outliers exist. The results shows that the system can better assist the users in discovering outliers and multiple trends.

Chapter 4

Nugget Browser: Visual Subgroup Mining and Statistical Significance Discovery in Multivariate Dataset

In this chapter, I present a novel pattern extraction and visualization system, called the Nugget Browser, that takes advantage of both data mining methods and interactive visual exploration. This work was published in IV 2011 [31].

4.1 Introduction

Subgroup discovery [7] is a method to discover interesting subgroups of individuals, such as "the subgroup of students who study in small public high schools who are significantly more likely to be accepted by the top 10 universities than students in the overall population". Subgroups are described by relations between independent (explaining) variables and a dependent (target) variable, as well as a certain interestingness measure. There are many application areas of subgroup discovery. For example, the extracted subgroups can be used for exploration and description, as well as understanding the relations between a target variable and a set of independent variables. Each subgroup or a set of subgroups is a pattern, i.e., a sub-region in the independent space. Detailed examination of such regions can be useful to improve understanding of the process that results in the pattern.

The subgroup discovery process poses many challenges:

- First, since the analysts may not know in advance what kind of interesting features the data contains, they may have to repeatedly re-submit queries and explore the results in multiple passes. For example, when the user submits a mining query, they need to specify the target attribute range of interest, such as the top 10 universities mentioned before. However, for different datasets and different application scenarios, the number of the top universities may be different, so they might have to try several times to find an appropriate range. This makes the mining process tedious

and inefficient. Thus, we need an interactive mining process that allows analysts to submit queries dynamically and explore the results in an interactive manner.

- Second, without visual support, users can only examine the mining results in text or tables. This makes it very hard to understand the relationships among different subgroups and how they are distributed in the feature space. Besides, when the user explores the mining results, the results are often in a descriptive or abstracted form, such as summaries of the sub-regions. However, the examination of the instances in the region is also very important for understanding the data point distribution. Thus, without a visualization of the mining results, users cannot build connections between the patterns and the instances.

- Finally, adjacent subgroups should be aggregated and clustered when they are of the same interesting type. For example, given there are two subgroups of students, both of which have significantly higher acceptance rates than the population, and they are adjacent to each other in one independent attribute, such as the groups with medium and high income. Then the two subgroups should be aggregated, and reported or treated as a whole subgroup. One benefit is that this aggregate representation is more compact, which provides users a smaller report list for easy examination. Another benefit is that the compact representation can be more efficiently stored in a file and loaded in computer memory. However, the clustered mining results generally tend to be multi-dimensional arbitrary-shaped regions, which are difficult to understand, report and visualize. Therefore, conveying the pattern in a compact, easily understandable, and visualizable form is desirable.

Focusing on these challenges, our main goal was to design a visual interface allowing users to interactively submit subgroup mining queries for discovering interesting patterns. Specifically, our system can accept mining queries dynamically, extract a set of hyper-box shaped regions called *Nuggets* for easy understandability and visualization, and allow users to navigate in multiple views for exploring the query results. While navigating in the spaces, users can specify which level of abstraction they prefer to view. Meanwhile, the linkages between the entities in different levels and the corresponding data points in the data space are highlighted.

The primary contributions of this work include:

- A novel subgroup mining system: we design a visual subgroup mining system where users can conduct a closed loop analysis involving both subgroup discovery and visual analysis into one coherent process.
- An understandable knowledge representation: we propose a strategy for representing the mining results in an understandable form. In addition to storage benefits, this representation is easy for analysts to understand, and can be directly displayed using common multivariate visualization approaches.

- A 4-level structure model: we designed a layered model that allows users to explore the data space at different levels of abstraction: instances, cells, nuggets, and clusters.
- Visual representation for the nugget space: for each level, we design a view in which users are able to explore and select items to visualize. The connections between the adjacent layers are shown based on the user's cursor position.
- We implemented the above techniques in an integrated system called *Nugget Browser* in XmdvTool [64], a freeware multivariate data visualization tool.
- Case studies suggest that our visualization techniques are effective in discovering patterns in multivariate datasets.
- We performed user studies to evaluate the visual representations of the mining results.

4.2 Visual Subgroup Mining and a Proposed 4-Level Model

In this section, we introduce the subgroup discovery problem and the mining process. As mentioned in Sec. 4.1, a subgroup discovery problem can be defined in three main features: subgroup description, a target variable, and an interestingness measure function.

A subgroup in a multivariate dataset is described as a sub-region in the independent attribute space, i.e., range selections on domains of independent variables. For example, "male Ph.D. students in a computer science department whose age is large (larger than 25)" is a subgroup with constraints in the 4 independent attribute space, i.e., *gender*, *degree program*, *department* and *age*. The sub-groups can be initialized by partitioning the independent attribute space. Given a multivariate dataset, pre-processing partitions the data space into small cells by binning each independent variable into several adjacent subranges, such as low, medium and high ranges. The number of bins for each dimension is defined by users. Users can select bin numbers initially based on the cardinality or application domain of the datasets, and then change the bin number according to the mining result, such as the number of empty and low density cells. Each cell is a description of one subgroup element.

For the target attribute, based on the application and the cardinality, it can be continuous or discrete. The quality functions are different for these two target attribute types.

As a standard quality function, Nugget Browser uses the classical binomial test to verify if the target share is significantly different in a subgroup. The z-score is calculated as:

$$\frac{p - p_0}{\sqrt{p_0(1-p_0)}}\sqrt{n}\sqrt{\frac{N}{N-n}}$$

This z-score quality function compares the target group share in the sub-group (p) with the share in its complementary subset. n and N are subgroup size and total population size. p_0 is the level of target share in the total population and $(p-p_0)$ means the difference between the target shares. For continuous target variables mean patterns, the quality function is similar, using mean and variance instead of share p and $p_0(1-p_0)$.

Users can submit queries on the target attribute to specify target range or a significant level to measure the interestingness of each group. The subgroups with high quality measures are query results, i.e., discovered patterns. Users can visually explore the extracted patterns and furthermore, can adjust the previous query and perform a new loop of query processing.

Intuitively, we use color to represent the mining result in the cell level. The cells (subgroups) are colored gray if their quality measure doesn't satisfy the significance level (usually 0.05). If the z-score is larger than zero and the p-value is less than 0.05, the cells are colored red. This means that the target attribute share or the average target attribute value are significantly larger than the population. Similarly, for the cells whose z-score is less than zero and the p-value is less than 0.05, the cells are colored blue. This means that the target attribute share or the average target attribute value are significantly lower than the population. In this work, we use different colors to represent different subgroup types.

A direct way to report the mining results is to return all the colored cells. Notice that the number of cells is exponential in the number of independent attributes. The query result can be very large, which makes it hard for the user to explore and understand. Specifically, a large set of unrelated cells may not be desired, because:

- Users may only care about large homogeneous regions (subgroups of the same type) rather than a set of unrelated cells.
- Users may want to know how many connected regions there are and what the sizes are.
- The result should be in a compact form for ease of understanding.

Towards these goals, we computationally extract two higher level abstractions of the mining result, i.e., the nugget level and the cluster level.

In the cluster level, we aggregate neighbor cells of the same type to form a cluster i.e., a connected region (Fig. 4.1 (a)). The clustering results can be used to answer questions such as how many connected regions there are and what the sizes (number of instances or cells) are. There are two benefits for the result in the cluster level besides to ease exploration. The first one is that the number of clusters can reveal the distribution of the mining result, such as a single continuous large cluster or a set of discontinuous small clusters scattered in the space. This can assist users to better understand how the independent attributes influence the target share. Second, since the subgroups of the same type are generally treated as a whole set, the same treatment can be applied to all individuals in one cluster rather than each single cell. Since users might be only

concerned with the large clusters, we can further filter out the small clusters, based on a user-specified threshold. This idea of clustering cells is similar to grid-based clustering and more benefits are discussed in[63, 1]. The difference is that we cluster the cells of the same type in terms of their interestingness based on the significance level for a target attribute, while most of the grid-based clustering techniques only consider the densities of each cell.

Although there are some benefits to representing the mining result at the cluster level, the largest problem is that the clusters are generally arbitrarily-shaped sub-regions in multi-dimensional space. This makes it very difficult for the users to understand the shape of a cluster and visually represent a cluster. To deal with these problems, we propose another level between the cell level and the cluster level, i.e., the nugget level. Specifically, we aggregate neighbor cells to form larger block-structured hyper-boxes for compact representation and easier perception. This aggregation of a set of adjacent cells is called a *nugget*. A nugget can be unambiguously specified and compactly stored by two cells, i.e., a starting cell and an ending cell, which are two corners of the corresponding hyper-box. A nugget has two important properties: *irreducibility* and *maximality*.

- *irreducibility*: any sub-region (subset) of a nugget, also in the cell form, is still of the user's interest and meets the interestingness measure function requirement.
- *maximality*: a nugget cannot be extended in any direction in any dimension to collect more cells to form a larger one.

The concepts of irreducibility and maximality were proposed by [8]. We extend this idea to a multi-dimensional space to generate a set of largest hyper-rectangular regions that satisfy the query.

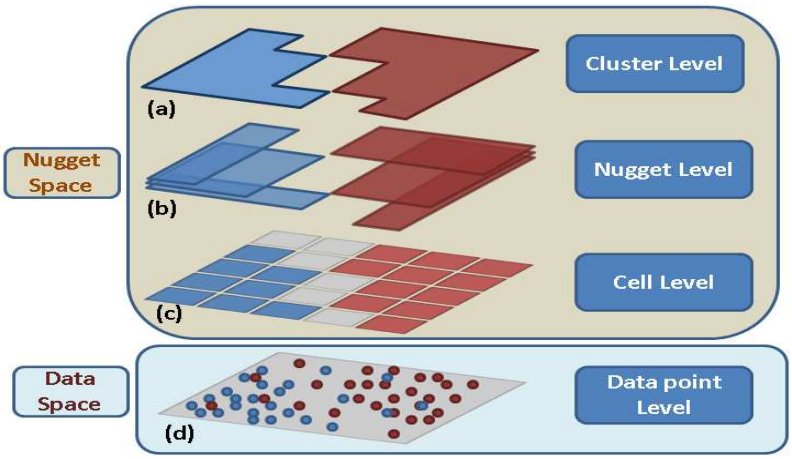

Figure 4.1: The 4-level layered Model. User can explore the data space in different levels in the nugget space.

The proposed 4-level structure model is shown Fig. 4.1. As shown in Fig. 4.1 (a), assume that the whole feature space is two dimensional (the gray plane) and the target dimension values (binary) are represented as the point color. In this example, assume the blue and red points are from two classes, e.g., USA cars and Japanese cars. Assume the user's query is requesting to find the subgroups where the target share (origin is USA) of the cars are significantly higher or lower than the population. To answer this, we first color the cells based on z-score: color the cell blue (red) if the percentage of cars from USA is significantly higher (lower) than the whole of the population. The partitioning and coloring results are shown in Fig. 4.1 (c). A gray cell means no significance is detected or are empty cells.

4.3 Nugget Extraction

In this section, we describe our proposed nugget representation and extraction method. Assume there are D dimensions in the feature space. As mentioned before, each dimension is partitioned into several bins. Assume there are B_k bins for dimension k. The cut points for dimension k are $C_{k,1}$ $(min) < C_{k,2} < \cdots < C_{k,B_k+1}$ (max). Here $C_{k,j}$ means the value of the j^{th} cut point in dimension k, assuming the first cut point is the minimum in this dimension. For any cell x, we assign an index (entry) based on its value position in each dimension: [$I_{x,1}$, $I_{x,2}$, $\cdots$, $I_{x,D}$] ($1 \leq I_{x,k} \leq B_k$, for $1 \leq k \leq D$). For example, if the first dimension value lies between the minimum and the second cut point, i.e., $C_{1,1} \leq v < C_{1,2}$, the index value of the first dimension of this instance is 1.

Definitions and the nugget extraction algorithm are introduced below:

Sort all cells: we define a cell c_a as *ahead of* another cell c_b if for a dimension k, $I_{c_a,k} < I_{c_b,k}$, and for the previous indices, they are all the same, i.e., $I_{c_a,t} = I_{c_b,t}$ for $1 \leq t < k$. We sort all the cells according to this order. We call the sorted list *CellList*. Some positions could be missing if the cell with that index is empty.

Of the same type: two cells are *of the same type* if they both satisfy the same query. This means they have the same color.

Previous cell: c_a is the *previous cell* of cell c_b in dimension k if $I_{c_a,k} = I_{c_b,k}$ - 1, and for the other indexes, they are the same, i.e., $I_{c_a,k} = I_{c_b,k}$ for $1 \leq j \leq D$ and $j \neq k$. So usually one cell has D *previous cells* in terms of all the dimensions.

Between two cells: cell c_x is *between* c_a and c_b if for each dimension, the index of c_x is larger than or equal to c_a, and smaller than or equal to c_b, i.e., $I_{c_a,k} \leq I_{c_x,k} \leq I_{c_b,k}$, for $1 \leq k \leq D$. If cell c_x is between c_a and c_b, it means c_x is covered by the hyper-box whose two corners are c_a and c_b. Note that here '*between*' does not mean the location in *CellList*.

Reachable: cell c_b is *reachable* from c_a if a) c_a and c_b are of the same type, and b) all the cells *between* these two cells are of the same type as c_a and c_b. If c_b is *reachable* by c_a, then that means the hyper-box, taking c_a and c_b as corners, is colored uniformly.

Algorithm Description: To find all the nuggets, for each cell c_x, we fill a list of cells, called *reachList*. If cell c_y is in the *reachList* of c_x, that means c_y is reachable from c_x. We fill this list from an empty list for each cell in the order in *CellList*. This is because

when filling the *reachList* for cell c_x, we have finished the lists of the D (maybe fewer) *previous cells* of c_x. Due to the property of *irreducibility*, we only examine the cells in the list of *previous cells* for filling the list for the current cell. After getting the union of all the *reachList*s of all the *previous* cells, we check each cell in the unioned list and delete unreachable cells. For this purging process, again only the previous cells' *reachList* require access. In order to fulfill the *maximality* property, those surviving cells, which can reach the current cell, have to be removed from the *reachlists* of the previous cells. The area between cell c_x and c_y (a cell in the *reachlists* of c_x) is a nugget.

The time cost for extracting all nugget is determined by the number of interesting cells. The number of all cells is independent of the data size and exponential to the dimensionality, given a fixed constant bin partition number for all dimensions. This means our system scales well as the data size increased and does not scale well as the dimensionality increases. However, due to the fact that most of the cells are not interesting (not colored), we can infer that the number of interesting cells for forming nuggets and clusters is low, which can lower down the time of extracting nuggets. To show this, we used our system to extract nuggets for two datasets with higher dimensionality. The first dataset has 13 dimensions and 73 nuggets were extracted in total; The second dataset has 30 dimensions and 18 nuggets were extracted in total. For both datasets, the time for extracting all nuggets cost less than 1 seconds, which indicated that this proposed nugget extraction process can handle datasets with high dimensionality.

4.4 Nugget Browser System

In this section, we introduce the system components, views, and the interactions.

4.4.1 Data Space

We employ Parallel Coordinates (PC), a common visualization method for multivariate datasets [40], to visualize the data points and nuggets. In parallel coordinates, each data point is drawn as a poly-line and each nugget is drawn as a colored translucent band (Fig. 4.6), whose boundaries indicate the values of the lower range (starting cell) and upper range (ending cell) for each dimension. The color blue and red indicate the sign of the z-score and darker color means higher significance is discovered for the subgroup. The color strategy is obtained from Color Brewer [16], using diverging color schema (7 bins). We provide interactions in the nugget navigation space view so that the users can select which data points to view in the cell, nugget and cluster level. The last dimension (axis) is the target attribute that guides the user in submitting queries and changing the target share ranges. The query ranges are shown during adjustment (vertical colored bars on the last axis). To assist the user in filtering out uninteresting nuggets, a brush interaction is provided. The user can submit a certain query range in the independent variable space and all the nuggets that don't fully fall in the query range will be hidden in the nugget view. An example of a query is to select all the subgroups within a certain age range.

4.4.2 Nugget Space

In the nugget space view, three coordinated views, i.e., cluster view, nugget view, and cell view, are shown in different 2D planes (Fig. 4.7). The linkages show the connections between adjacent views [15].

Cluster View. In the cluster view (Fig. 4.7 left), we employ a small "thumbnail" of a parallel coordinate view to represent each cluster. The size of each thumbnail is proportional to the number of instances each cluster contains, so that large clusters attract the user's attention. When the user moves the cursor onto a cluster, the parallel coordinate icon is enlarged and the connections are shown from this cluster to all the nuggets in the nugget view that comprise this cluster. Meanwhile, the corresponding instances are shown in the data space view.

Since the clusters consist of the data points in a high-dimensional space, to preserve the high-dimensional distances among the clusters we employ an MDS layout [12] to reveal latent patterns. The question is how to measure the similarity of two clusters. A commonly used and relatively accurate method for measuring the distance between two groups of instances is to average all the Euclidean distances of each instance pair from different groups. The problem is that for large clusters, the computational cost is high. We therefore calculate the distance in a upper level of the proposed 4-level model, i.e., using the average Euclidean distances between all cell pairs. As a result, the cost reduces as it depends on the number of cells, which is much smaller. The cell distance is calculated as the Euclidean distance between two cell centroids.

Nugget View. As mentioned before, each nugget is a hyper-rectangular shape. A single star glyph with a band, as proposed in [68], can thus be used to represent a nugget (Fig. 4.7 middle). The star glyph lines show the center of the nugget, and the band fades from the center to the boundaries. Similar to the cluster view, connections between the nugget view and the cell view are displayed according to the user's cursor position. The corresponding data points are also highlighted.

We again use an MDS layout for the nugget view, but the distance metrics are calculated differently from the cluster view. This is because any two nuggets could overlap in space, thus an instance could be covered by multiple nuggets. To reveal the distance between two nuggets, we designed two different distance measurements: one for overlapping nuggets and one for non-overlapping nuggets.

When the two nuggets have common cells, the distance metric indicates how much they overlap:

$$Dis(Nugget_A, Nugget_B) = \frac{|A| + |B| - 2|A \cap B|}{|A| + |B|}$$

Here $|A|$ means the number of cells that cluster A includes. When the two cells have a very small overlapping area, i.e., almost non-overlap, the distance is near 1. When the two cells almost fully overlap on each other, the distance is near 0.

When the two nuggets do not have any common cells, we use the Manhattan distance as the measurement. For each dimension, the distance is measured by using a grid as a

single unit, called the *grid distance*. For example, the grid distance for dimension k is 0 if on that dimension the two nuggets' boundaries meet without any gaps, or the two nuggets have overlapping bins (note that two nuggets may not overlap in space, but may overlap in certain dimensions). The grid distance of dimension k is 1 if there is a one-bin gap between the two nuggets on that dimension. The distance in any dimension is the cell distance $+1$ indicating how many steps they are away from each other:

$$Dis(Nugget_A, Nugget_B) = \sum_{k=1}^{D} (GridDistance_k(A, B) + 1)$$

Note that the minimal distance is 1 for two non-overlapping nuggets, which is also the maximal distance for two overlapping nuggets. Hence in the MDS layout view, the nuggets in a cluster will tend to stay together to help reveal patterns.

For both of the cluster view and the nugget view, we use MDS as the layout strategy, so the time cost for constructing the two views is mainly determined by the MDS algorithm. Here we give the time cost of two datasets for constructing this view to show how our system scales as the number of nuggets increases. For the first dataset with 69 clusters and 86 nuggets, it took 5 seconds to create this view; and for the second dataset with 123 clusters and 248 nuggets, it took 155 seconds to creating this view. This indicated that as the number of clusters and nuggets increase, our system doesn't scale very well. To address this limitation, implementing a faster algorithm of MDS layout can be a future work.

Cell View. In the cell view (Fig. 4.7 right), each cell is represented as a square. The cell colors are consistent with the colors in other views. The cell is highlighted when the user is hovering the cursor on it. Meanwhile, all the data points in this cell are shown in the data space view. The curves indicating connections between the cell level and the nugget level are also shown for the cells the cursor points to. Instead of a single curve, multiple ones are shown as a cell could be included in multiple nuggets.

4.5 Case Study

In this section, we discuss a case study showing the effectiveness of our system. The dataset was obtained from the UCI Machine Learning Repository called "the Mammographic Mass Dataset" [60]. Mammography is the most effective method for breast cancer screening. The dataset size is 961 (830 after removing instances with missing values). 5 independent attributes, such as the *age* of the patient and the *density* of the mass, are extracted and the target attribute is *Severity* (benign or malignant). There are two main goals for analyzing the dataset. The first one is to understand how the independent attributes influence the target attribute. This can assist the doctors in finding the important attributes impacting the diagnosis results. The second goal is to discover the subgroups where the benign (malignant) rate is significantly higher or lower than the population. For a future diagnosis, if a patient is discovered in those groups, more attention should be paid or some conclusion about the diagnosis result could be drawn.

To show the difficulty of finding how the independent attributes influence the target attribute using common multivariate data visualization techniques and interactions, we first display the dataset using Parallel Coordinates in XmdvTool. As shown in figure 4.2 and 4.3, the highlighted instances are selected using the brush technique (range query) on the target attribute. Figure 4.2 shows the query result on all the benign instances (red color poly-lines) and figure 4.3 shows the query result on all the malignant instances. The pink area shows the bounding box of all the instances in the query. It can be observed that for each query, the instances cover almost the whole attribute ranges and all different values in different dimensions. This shows the common visualization technique, even with interactive range queries, can hardly reveal the relationship between the independent attributes and the target attribute.

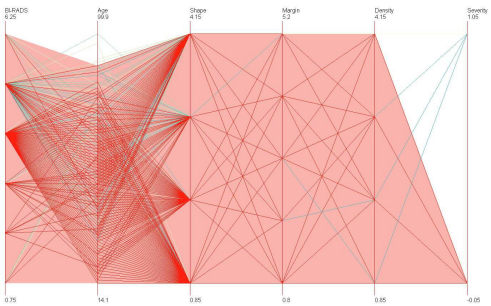

Figure 4.2: Brushed benign instances

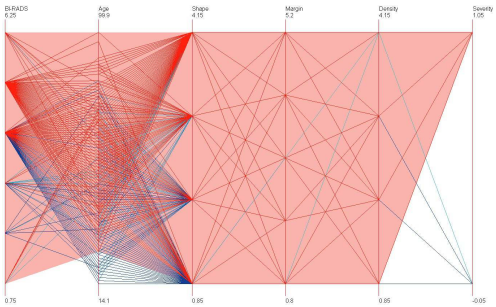

Figure 4.3: Brushed malignant instances

We then show the insufficiency of the traditional subgroup mining technique without visualization in providing compact and easily understandable mining results. We performed the mining as follows. The target share value is benign in the target attribute. This query examines the subgroups with significantly higher benign rate and significantly lower benign rate. Note that significantly lower benign rate does not necessarily mean significantly higher malignant rate, which can be examined by specifying another mining query that takes share value as malignant in the target attribute. The whole independent attribute space is portioned by binning each attribute. Specifically, for the attribute whose cardinality is smaller than 7, the bin number is the same as the cardinality, such as *density*. For the attribute *age* (numerical attribute), the bin number is set to 7. We chose 7 because for lower values, the patterns are very similar, but less clear, while higher number of bins results in a lower number of instances in each group, which reduces the reliability of significance due to the small sample size. After the binning, the whole dataset is partitioned into a set of subgroups. Each subgroup consists of a group of individuals whose attribute values are similar or the same in all dimensions. Each subgroup is examined using the p-value and z-score of the statistical test as the interestingness measure. Parts of the mining results are shown in Figure 4.4 as a table. The star means the description of each subgroup in each dimension. 18 subgroups have the benign rate significantly larger than the population. It is clear that without visualization, analysts cannot understand how the subgroups are distributed in the space and the relationships between the subgroups.

Also, for some subgroups, such as number 12, 13, and 14, they are adjacent to each other and can be reported as a single group for a compact representation.

	BI-RADS						Age							Shape				Margin					Density					
Group #	1	2	3	4	5	6	1	2	3	4	5	6	7	1	2	3	4	1	2	3	4	5	1	2	3	4	z_score	p_value
1			*						*					*				*							*		1.683	0.047
2				*			*							*				*							*		3.385	0.001
3				*			*									*		*							*		2.385	0.008
4				*			*							*				*							*		1.683	0.046
5				*				*						*				*							*		5.125	0.001
6				*				*							*				*					*			1.683	0.046
7				*				*							*			*							*		3.814	0.001
8				*				*							*					*					*		2.177	0.001
9				*					*						*				*					*			1.683	0.046
10				*					*						*				*						*		5.642	0.001
11				*					*						*				*					*			1.945	0.026
12				*					*						*			*							*		4.069	0.001
13				*					*						*				*						*		1.945	0.026
14				*					*						*					*					*		1.683	0.046
15				*							*				*			*							*		5.186	0.001
16				*						*					*			*						*			1.945	0.026
17				*						*					*			*							*		4.562	0.001
18				*							*				*			*							*		2.582	0.004

Figure 4.4: The mining results are represented in a table before aggregating neighbor subgroups.

From the previous discussions, we can observe several difficulties:

- it is hard to understand how the independent attributes influence the target attribute using common visualization techniques.
- it is hard to understand the distribution of the subgroups
- the mining results are not reported in a compact knowledge representation form.

Next we will show how to use the Nugget Browser system to better solve the subgroup mining problem. Figure 4.5 shows the higher level, i.e., the nugget level representation of the mining result in a table form. 8 nuggets are reported in a more compact manner, compared to the result of traditional subgroup mining, i.e., a list of subgroups. Figure 4.6 shows all the nuggets (translucent bands) extracted in the data space view. Color blue means a significantly higher benign rate and color red means a significantly lower benign rate. It is very clear that subgroups with high benign rates can be differentiated from the low benign rate subgroups in most of the dimensions, which indicates that the independent attributes have a strong impact on the target attribute. However, this influence can hardly be discovered in traditional multivariate data visualization techniques, even with range queries. Specifically, the high benign rate subgroups have lower values for attributes *BI-RADS*, *Age*, *Shape* and *Margin*, compared to the low benign rate subgroups. Most of the subgroups with significance discovered have *Density* value 3 (means low). More details of how the independent attributes influence the target attribute will be discussed later.

	BI-RADS						Age							Shape				Margin					Density			
Nugget #	1	2	3	4	5	6	1	2	3	4	5	6	7	1	2	3	4	1	2	3	4	5	1	2	3	4
1			*	*					*					*				*								*	
2				*			*	*						*	*			*						*	*	
3				*					*						*			*	*	*					*	
4				*			*	*	*	*	*			*	*			*							*	
5				*			*	*	*	*	*				*			*							*	
6				*				*	*	*					*			*						*	*	
7				*				*	*						*					*					*	
8				*			*							*	*	*		*							*	

Figure 4.5: The mining results are represented in a table after aggregating neighbour subgroups.

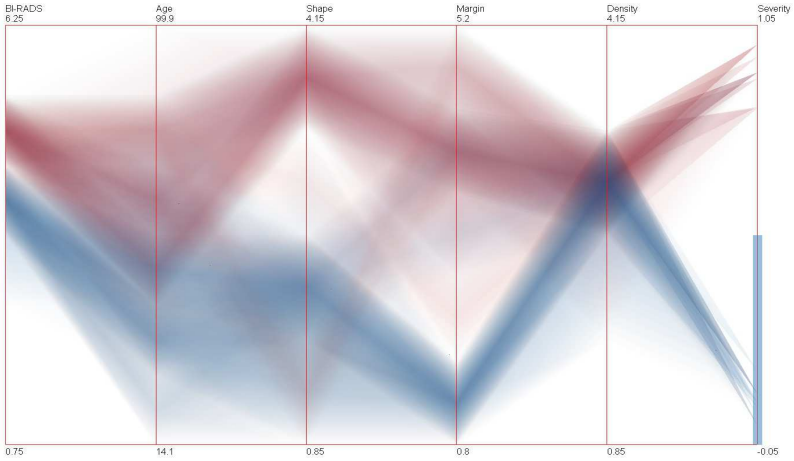

Figure 4.6: The data space view shows all the nuggets as the translucent bands. The rightmost dimension is the target attribute. The blue vertical region on the target dimension indicates the target range of the subgroup mining query.

Although the nugget representation, shown in Figure 4.5, is more compact than the cell representation, without the visual representation, the users still have difficulties understanding the distribution of the nuggets and building connections between the pattern and the instances. To better understand the mining results and further explore them, the analysts can open the nugget space view (Figure 4.7). Based on the distribution in the nugget view and the cluster view, the high benign rate cluster and the low benign rate cluster are separated from each other in the attribute space, indicating that the the target attribute is influenced by the independent attributes. We can also discover that a large red cluster and a large blue cluster are extracted. It is shown that the higher benign rate regions and low benign rate regions are continuous in the independent attribute space. More discoveries found during the exploration in the nugget space are as follows:

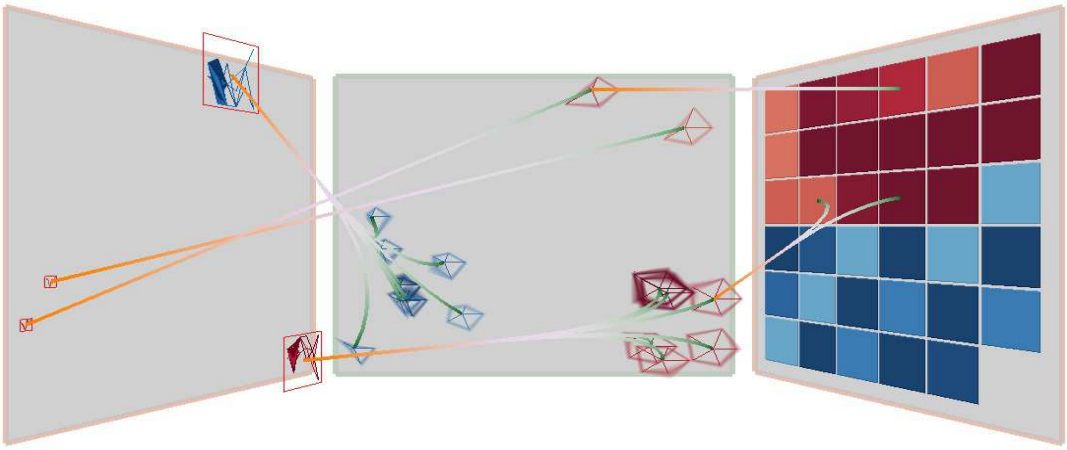

Figure 4.7: The nugget space view shows the mining result in 3 level of abstractions. The connecting curves indicate the connection between adjacent levels.

1. For the low benign rate subgroups, there are two outliers outside the main cluster. By hovering the cursor and selecting on the two outliers, we can discover what causes the two outliers to differ from the main cluster: the *Shape* values of the main cluster (red) are 3 and 4, while the two outliers have *Shape* value 1. When showing these two outlier subgroup instances in the data space view, we can observe that no instances are benign and the group sizes are small. Thus, the doctors can consider that they are not typical and ignore these two outlier subgroups during analysis.

2. The shape value 4 is more important for the low benign rate. This can be discovered when displaying all the instances in the red cluster: the shape values are either 3 (means lobular) or 4 (means irregular), while for the value 4 , higher significance is found, which can be recognized by a darker color.

3. For lower age patients, higher benign rate tend to be discovered. This can be verified by distribution of the interesting subgroups: no higher benign rate groups are in age bin 6 and 7; no lower benign rage groups are in age bin 1 and 2.

4. Attribute *BI-RADS* has a negative effect for higher benign rate, i.e., lower *BI-RADS* values tend to have higher benign rate. This can be discovered according to the distribution of subgroups with significance on this attribute. For the higher benign rate subgroups most of them have *BI-RADS* value 4. For low benign rate subgroup: most of them have *BI-RADS* value 5. The analysts can understand this trend better if they know the meaning of this attribute: each instance has an associated BI-RADS assessment. The lowest value means definitely benign and highest value means highly suggestive of malignancy.

4.6 User Study

In this section, we discuss a user study for evaluating the effectiveness of the visual representations of the nugget space. The hypothesis is that compared to the current existing work using a table representation, the designed nugget view that uses shape and layout can better help the users understand the subgroup mining results and quickly identify patterns in it. The patterns could be outliers, clusters, and overlapped subgroups. To show that simple multivariate visualization techniques cannot reveal the pattern very well, we implemented scatterplot matrices for comparison. Another goal of this user study is to evaluate the proposed 3-level knowledge representation. We compared the nugget view and the cluster view, in terms of their abilities of prediction. For this task, the subjects were provided with the nugget view, or the cluster view, as well as a set of instances. They were asked to build a correlation between the subgroups and the instances.

We invited students as the subjects (18 in total) to participate in the user study. The subjects were asked to answer 11 questions based on different visual presentations of the subgroups. The subjects answered the questions based on screen-copied figures which were printed out on paper. Note that any single question about the subgroup mining results could be answered based on different visual representation methods, such as the designed nugget representation or the table representation. Subjects were randomly assigned a visual representation method to answer a given question. Take the evaluation of the representation of the two levels (nugget level and cluster level) for example. We designed two questions (question Q_a and question Q_b) to compare the representation of the two levels. We generated two groups of questions, group G_A and group G_B, as follows. Each question group had both questions Q_a and Q_b. In group G_A, question Q_a would be answered based on the nugget level representation, while question Q_b would be answered based on the cluster level representation. In group G_B, the questions are the same, but we exchanged the visual representations: question Q_a was represented using the cluster level and question Q_b was represented using the nugget level. In the study, we randomly assigned half of the subjects to question group G_A and the other half to question group G_B. Similarly, we generated three groups of questions to evaluate the three representations of the subgroup mining result: table representation, scatterplot matrix representation, and the nugget representation.

Before the study, the subjects signed a consent form. Then each subject was shown a brief explanation of the study using examples and sample questions, such as which dataset we used and how to read the figures. The subjects finished the study by answering several questions. One type of question was the identification task, i.e., identify and highlight the specified pattern, such as clusters and outliers. This type of questions is used to evaluate the visual representation of the subgroup mining result. The other type of question was the prediction task, i.e., for a given instance, determine which subgroup it belongs to. This type of questions was used to evaluate the nugget representation in different levels. We recorded the time each subject spent on each question for further analysis.

Figure 4.8 uses error bars with a 0.95 confidence interval to show the accuracy for the three knowledge representations of the subgroup mining results. We found that the

scatterplot data representation and the table representation were very similar in terms of accuracy. It is clear that the proposed nugget representation is better than both the scatterplot and table representation, though no statistical significance is detected.

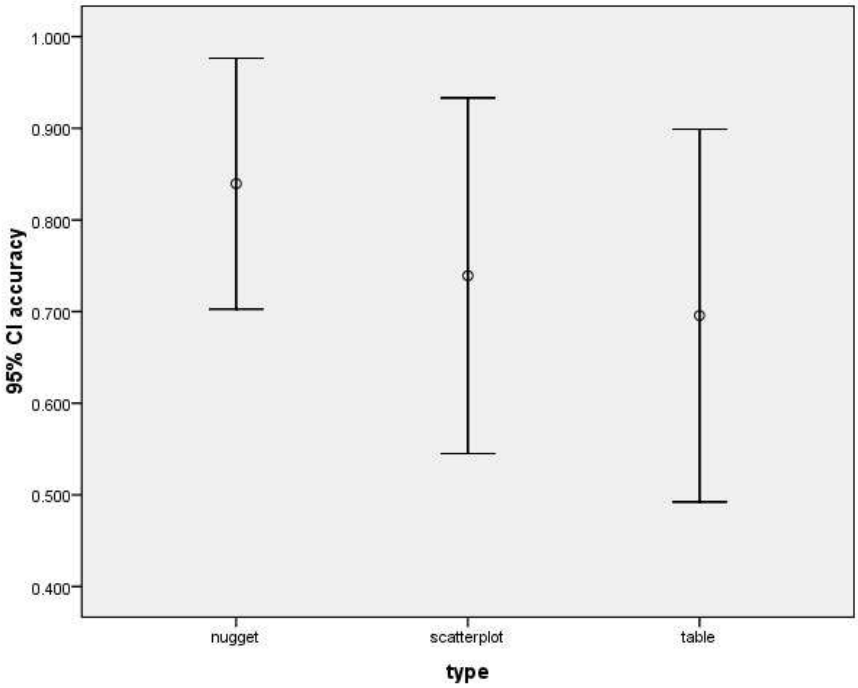

Figure 4.8: The comparison of accuracy for different mining result representation types.

We also examined time spent on each representation, which are shown in Figure 4.9. Similarly, the nugget representation is better than the other two types of knowledge representations. The difference between the nugget representation and the scatterplot matrix is significant (p-value=0.014). To conclude, we found the visual representation of the subgroup mining result using the proposed nugget method is better than the tabular method and scatterplot matrix method, in both accuracy and time.

Lastly, we compared the two different levels of nugget representations. Figure 4.10 compares the accuracy for these two levels, and Figure 4.11 compares the time spent for these two levels. It is shown that the two representations have their own advantages: the nugget representation costs less time for the prediction task, while the cluster level can provide higher accuracy. A possible explanation is that the cluster view uses the parallel coordinates thumbnail representation, which may cause overlapping when a cluster has many instances. This explains why the subjects spent more time on this task when using the cluster view. The accuracy for the nugget view is not as good as the cluster view. This is because compared to the star glyph, a parallel coordinate representation is relatively easier to make a comparison, as a trend is shown. Since no statistical difference is detected, the conclusion is that there is no difference in the two methods in terms of their prediction abilities.

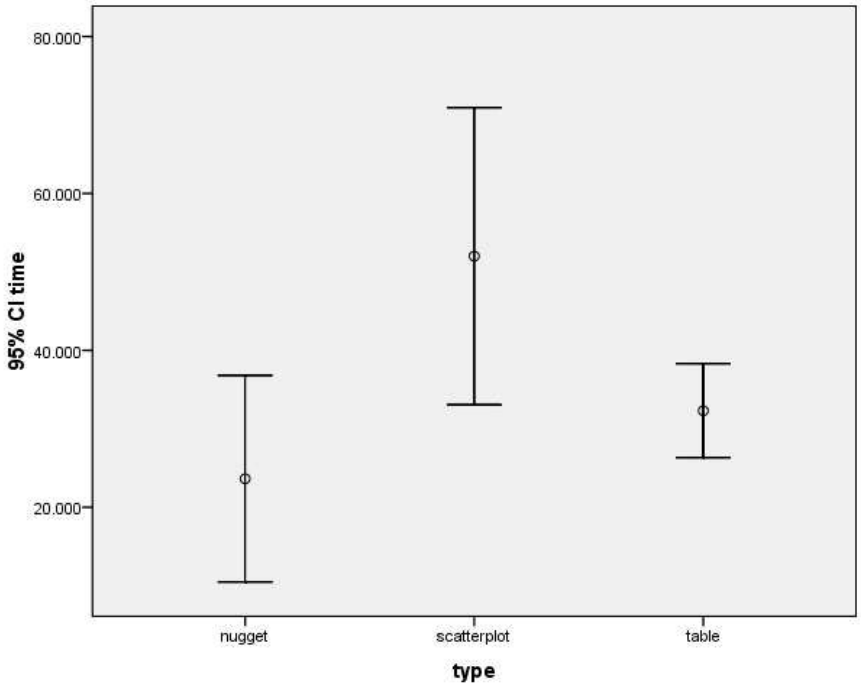

Figure 4.9: The comparison of time for different mining result representation types.

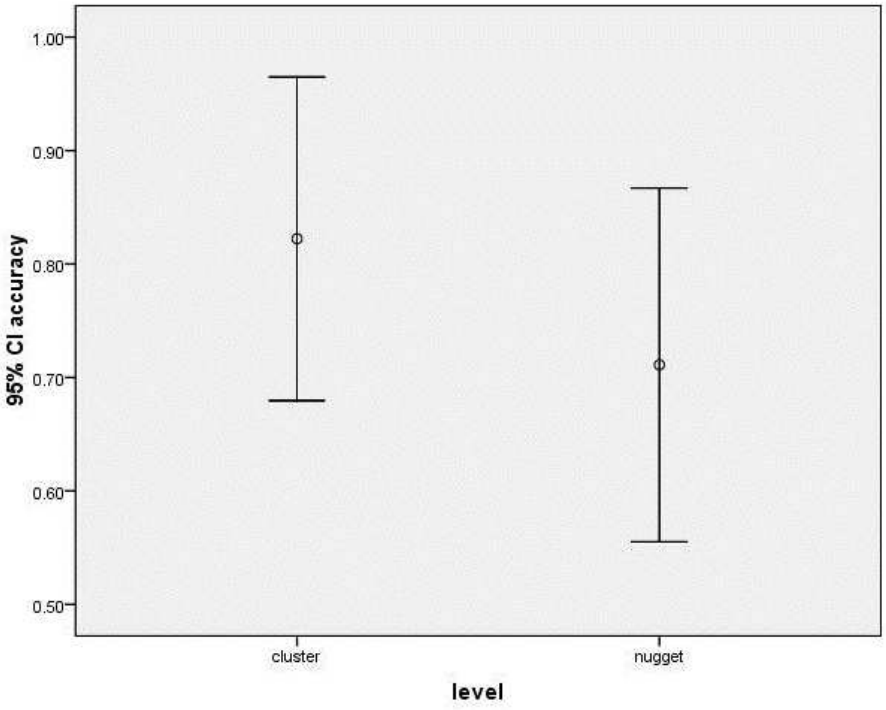

Figure 4.10: The comparison of accuracy for different levels.

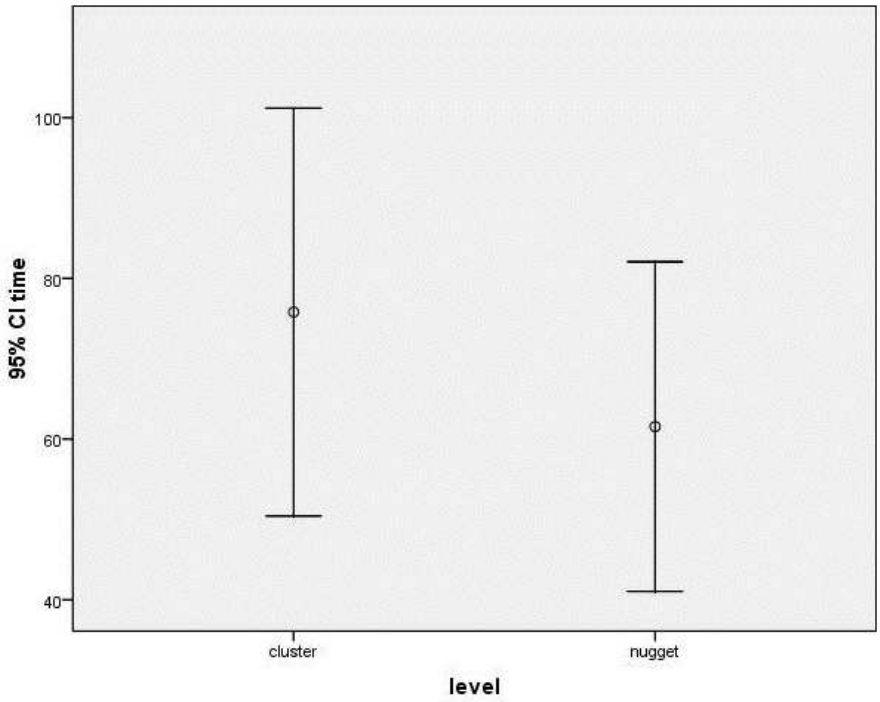

Figure 4.11: The comparison of time for different levels.

Conclusions

In this chapter, we described a visual subgroup mining system, called the *Nugget Browser*, to support users in discovering important patterns in multivariate datasets. We proposed a 4-level layered model that allows users to explore the mining result in different levels of abstraction. The nugget level mining results are represented as regular hyper-box shaped regions, which can be easily understood, visualized, as well as compactly stored. The layout strategies help users understand the relationships among extracted patterns. Interactions are supported in multiple related nugget space views to help analysts navigate and explore. The case studies show how our system can be used to reveal patterns and solve real life application problems. The user study shows that the proposed visual representation can better help the analysts understand the subgroup mining results, as well as quickly identify specified patterns.

Chapter 5

Local Pattern and Anomaly Detection

In this chapter, I introduce a novel visualization system that allows analysts to perform multivariate analysis in a pointwise manner and examine anomaly patterns in multivariate datasets. This system is designed for one type of local analysis, i.e., sensitivity analysis. I evaluated the system with formal user studies and expert studies. There are two main goals when using this system to explore a multivariate dataset:

- examine a multivariate dataset from a single focal point. The relationships between neighbors and the focal point are visually represented.
- discover anomalous local patterns, i.e., outliers that are different from the global pattern. Each detected anomalous local pattern can be viewed as a *nugget* that can be managed using this system.

5.1 Introduction

5.1.1 Sensitivity Analysis

A local pattern can be viewed as a pattern that is extracted only from the subset of data points within a small region around a focal point. There are many multivariate analysis techniques that follow the idea of local analysis, and one of them is sensitivity analysis. Sensitivity analysis is the study of the variation of the output of a model as the input of the model changes [56]. When we study the correlation between a target (response) variable Y and a set of independent variables $\{X_1, X_2, \ldots, X_n\}$, sensitivity analysis can tell analysts the change rate of Y as X_i varies. Analysts can also discover which input parameters are significant for influencing the output variable. Sensitivity analysis has been widely applied for understanding multivariate behavior and model construction for analyzing quantitative relationships among variables [56]. For example, it can be applied to car engine designs: fuel consumption is dependent on the relationships among the design choices, such as fuel injection timing, as well as operation-varied conditions, such as engine speed [42]. The analysis results are important in helping engineers tune the parameters in designing an engine.

Sensitivity analysis is essential for decision making, system understanding, as well as model constructing. Numerous approaches have been proposed to calculate the sensitivity coefficients. I focus on differential analysis, where sensitivities are defined as the partial derivatives of a target variable with respect to a set of independent variables.

5.1.2 Motivations for Pointwise Exploration

Although many visual analytics systems for sensitivity analysis follow this local analysis method, there are few that allow analysts to explore the local pattern in a pointwise manner, i.e., the relationship between a focal point and its neighbors is generally not visually conveyed. The key idea behind this research is analogous to the street view in a Google map [29], where the user can stand in a position in the global map to browse the neighbors and understand its vicinity, i.e., the local patterns.

This pointwise exploration is helpful when a user wants to understand the relationship between the focal point and its neighbors, such as the distances and directions. The analysis result can assist analysts in understanding which neighbors do not conform to the local pattern. This discovery can be used to detect local anomalies and find potentially interesting neighbors.

To better understand the usefulness of pointwise sensitivity analysis, I discuss an application scenario for selecting an apartment near a campus. The target variable is the price and the independent variables are several apartment attributes that influence the target, such as room size, bedroom number, distance to campus, and so on. The local sensitivity analysis can tell users (students) how the price is influenced by an independent variable, either positively or negatively, as well as which variables are important for choosing an apartment. However, users often cannot easily decide which apartment is worth renting. Given a particular apartment or the one in which they currently reside, it is not always clear whether there are any better choices compared to this one. Specifically, can the student pay a little more to get a much better apartment, or find a similar one that is much cheaper. Finally, if users have domain knowledge or certain requirements, they should be able to use this to change this apartment finding task. For example, if the students know that distance is much more important for their choices, i.e., they prefer a closer one rather than a bigger one (assume both choices increase costs the same amount), they should increase the influencing factor for distance, or similarly decrease the influencing factor of size.

We seek to develop a system focusing on the challenges that mentioned in the apartment finding problem. In this chapter, I present a novel pointwise local pattern visual exploration method that can be used for sensitivity analysis and, as a general exploration method, for studying any local patterns of multidimensional data. Specifically, this system allows users to interactively select any single data instance for browsing the local patterns. Each instance is assigned a factor using statistical means to reveal outliers that do not conform to the global distribution. In the local pattern view, the layout strategy reveals the relationships between the focal point and its neighbors, in terms of the sensitivity weighting factors. Users can interactively change the sensitivity information, i.e.,

the partial derivative coefficients, based on their requirements. Users can also compare the local pattern with the global pattern both visually and statistically.

The primary contributions of this system include:

- *A novel pointwise exploration environment*: It supports users in browsing a multivariate dataset from a *pointwise* perspective view. This exploration assists users in understanding the vicinity of a focal point and reveals the relationships between the focal point and its neighbors.
- *A novel visualization approach for sensitivity analysis*: Sensitivity analysis is one important local analysis method, thus is well suited for our pointwise exploration. The designed local pattern exploration view indicates the relationships between the focal point and its neighbors, and whether the relationship conforms to the local pattern or not. This helps the user find potentially interesting neighbors around the focal point, and thus acts as a recommendation system.
- *Adjustable sensitivity*: The system allows users to interactively adjust the sensitivity coefficients, which gives users flexibility to customize their local patterns based on their domain knowledge and goals.
- *System evaluation using real-world dataset*: I evaluated the effectiveness of our system based on a real-world dataset and performed a formal user study to better evaluate the effectiveness of the whole framework.

5.2 Local Pattern Extraction

5.2.1 Types of Local Patterns

There are many types of local patterns that can be used for sensitivity analysis and other local analysis. I list some of them in the following:

- the model coefficients: one way to calculate the sensitivities is to use the partial derivative values. For a local linear regression model, the coefficients for each independent variable are one of type local pattern. The size of this pattern is the number of independent variables.
- the residual values: for each neighbor, the residual describes how well it fit the local model, i.e., the difference between the estimated value and observed value. The size of this pattern is the neighbor count.
- the angle between the connecting vectors, i.e., a vector from the focal point and a neighbor, as well as a certain orientation, such as the norm vector of the linear regression hyper-plane, or a positive direction of a dimension. I use this information as the local pattern and it will be discussed later in Section 5.2.

5.2.2 Neighbor Definition

A local pattern in a multivariate dataset means a single data point and its vicinity, i.e., a set of data items in a sub-region. The data point can be viewed as a focal point F, which could be an existing data item in the dataset or any user-specified position. The data items within the range are considered neighbors.

The notion of locality can be defined by the user's requirement. Generally, two different types of 'locality' are considered:

- *Hyper-sphere*: The definition of a Hyper-spherical local neighborhood with respect to F is a subset of points within a hyper-sphere taking F as the center. This set of normalized n-dimensional points is denoted as V, with $|F - v| \leq r$ ($v \in V$). $|F - v|$ means the distance measure (Euclidean distance) between the focal point F and the point v. r is a user-defined radius to control the degree of locality. Note that the normalization means different weights will be assigned when calculating the *Euclidean* distance of the two data points. Usually all the dimensions are normalized between zero and one; however, in some cases, a weight can be assigned based on the model coefficients or any user-specified values. This neighbor definition is widely used in many multivariate analysis methods such as density based clustering [24]. Another way to define neighbors is to specify a number k, and for a specific point, the k nearest data points are its neighbors. In this case, the local region is also a sphere shaped area.

- *Hyper-box*: The definition of a Hyper-box local neighborhood with respect to F is a subset of points within a Hyper-box taking F as the center. This set of n-dimensional points v satisfy $|F_k - v_k| \leq r_k$, where F_k is the value of dimension k of F and r_k is the range of the hyper-box for dimension k. For a box-shaped area, the user can specify the box size on each dimension. This gives users flexibility to define the neighborhood based on different applications and requirements. For example, for categorical independent attributes, such as the country of origin or manufacturer of a car, the coefficients of the sensitivity analysis are meaningless, since the attribute values are not ordinal. However, for different *origins* or *manufacturers*, the coefficients may be different and it is useful to compare them. In this case, the user can specify the box size on the categorical attributes so that the cars of the same origin and manufacturer are neighbors. This system allows users to perform this neighborhood definition in a parallel coordinate view by dragging and resizing a box-shaped region.

5.2.3 Calculating Local Patterns for Sensitivity Analysis

As mentioned earlier, there are many ways to compute the sensitivity of one dependent variable with respect to an independent variable. In this work, I follow a variational approach, where the sensitivity can be calculated by the partial derivative of one variable with respect to another. The derivative of a target variable, y, as the independent variable,

x, changes is approximated as $\partial y/\partial x$. The relationship is geometrically interpreted as a local slope of the function of y(x). Since we do not know the closed form of the function y(x) between variables in the general case, we approximate the partial derivatives using linear regression. The regression analysis is performed in different neighborhoods around each point. A tangent hyperplane for each focus point is calculated based on its neighbors using linear regression. This linear function represents how the independent variables influence the target variable, considering a constant local changing rate for all independent variables. Also, the representation enables the model to predict the target value given the independent variables, as well as to assess the error between the predicted value and the observed value. In a sense, analysts assume that the local neighbors fit this trend since the sum of the square errors to the regression line is minimized.

Generally speaking, any local information that can assist analysts in performing local pattern analysis can be extracted and visually represented for examination, such as neighbor count, distances to neighbors, and orientation to neighbors. In this research, in particular, I focus on the orientations from the focus point to the neighbors. I choose this pattern for two reasons. First, this pattern tells users the relationships between the focus point and its neighbors, i.e., the directions to move from the focus point to its neighbors. Second, and more importantly, since our system is designed for sensitivity analysis and we extract a linear regression model, this direction reveals whether the relationship conforms with the local trend or not, which can assist analysts in performing sensitivity analysis in this neighborhood region.

Similar to the street view in Google Map, when a user stands at a single point (the focal point) to examine the neighborhood, the orientations to the neighbors tell users which direction they should move from the standing point (the origin) to reach each of the neighbors. In the map coordinate system, this direction is usually described using an angle between a standard direction vector, such as north, and a connecting vector, from the focal point to a neighbor point. In our system, to assist users in performing sensitivity analysis, we take the normal vector of the regression hyperplane as the standard direction. Since there are two normal vectors of one plane, without any loss of generality, we take the one directed to the negative side of the target variable as the standard normal direction. For each neighbor of the focal point, we calculate an angle θ between the normal vector of the regression hyperplane and the connecting vector between the focal point and that neighbor, as shown in Figure 5.1. $Cos(\theta)$ is the dot product of the two unit vectors.

To remove the unit differences among the different attributes, we assign a weight, using the regression coefficient, for each independent attribute, so that the changing rates are the same between each independent variable and the target variable. This step can be considered a normalization. After the normalization, the slopes of the linear trend are all π/4 in all dimensions, and the angle θ is between 0 and π. The direction of the normal vector is orthogonal to the local gradient, taking the focal point as the starting position. Therefore, the angle θ for one neighbor represents whether the relationship between the focal point and this neighbor conforms with the local linear trend or not. The expectation of this angle is $\pi/2$, assuming all the local points fit the extracted linear model very well. If the angle is $\pi/2$, it means that the vector from the focal point to this neighbor is the

same as the local trend (the blue point in Fig. 5.1). If the angle is less than $\pi/2$ (the green point in Fig. 5.1), it indicates that the neighbor's target attribute value is smaller than the estimate using the extracted model. Note that when we say the predicted value, we do not mean it is the predicted value using the local regression plane (the solid red line in Fig. 5.1). Since we care about the relationships between the focus point and its neighbors, the predicted value is based on the regression plane that is moved to the focal point in parallel (the dotted red line in Fig. 5.1). In contrast, if the angle is larger than $\pi/2$ (the yellow point in Fig. 5.1), it means that the neighbor's target attribute value is larger than the estimate, taking the origin as the focal point.

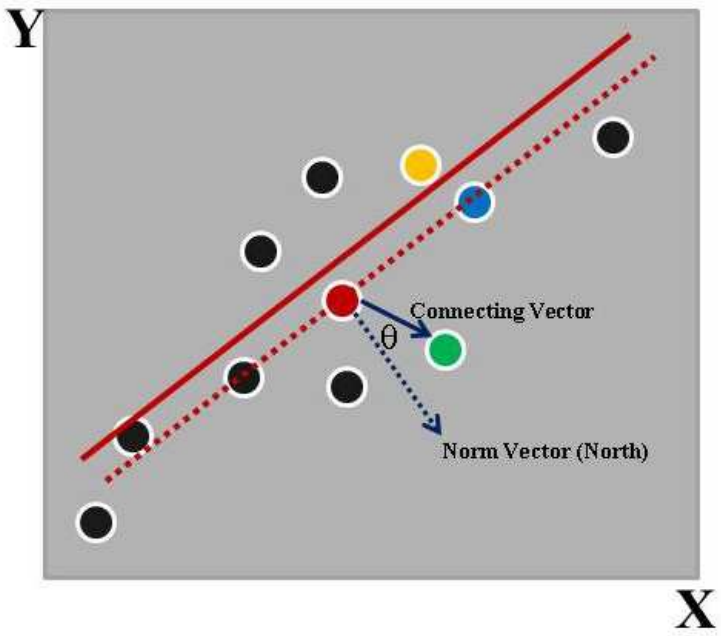

Figure 5.1: The extracted local pattern.

To sum up, in this system, the extracted local pattern for a single point is a vector V, in which each value is an angle introduced as before. The size of V is the same as the neighbor count.

5.2.4 Anomaly Detection

Our system allows users to detect anomalous local patterns that deviate from others. In general, we follow the idea of subgroup discovery mentioned in Chapter 4 to identify interesting subgroups from the dataset.

Since each local pattern is extracted from a small subset, i.e., neighbors of a single data point, we can take each local pattern as a subgroup. Thus subgroup discovery can be applied to discover the local patterns of certain special significance, such as the ones

different from the others, i.e. anomalies. The word "anomalous" implies that there is something basic to which each subgroup can be compared, i.e., there is some notion of 'background' or 'expected' pattern. For example, the directions (angles) to the neighbors mentioned before are expected to be $\pi/2$. We know this is because the analysts have knowledge of regression analysis. In general, however, users do not have this prior knowledge.

As a general solution, I assume each subgroup (a local pattern) is one extracted sample. All the samples could be integrated as a population to simulate the underlying model that generates the individuals. I use the term "global pattern" to represent the integrated pattern. Each local pattern is compared with this global one to decide whether it is different from it. To better understand this idea, I give an example for searching for anomalous patterns on a map. In this example, the extracted pattern is the percentage of water coverage around each sample position, and the goal is to detect anomalous areas in terms of this pattern. Since we assume users do not know the expected pattern, we integrate all the local patterns (percentages of water coverage) together and use a statistical test to detect anomalies. It is not hard to understand that for a map of mainland, areas near lakes and shores are anomalies; for a map of the ocean, islands are anomalies.

As a statistical method, the significance value of each local pattern is evaluated by a quality function. I use the same quality function mentioned in Chapter 4. Although this is a general way to detect anomalies, visual exploration on each single pattern is still often needed. This is because this approach is based on the assumption that the population is normally distributed, which does not always hold for all applications. In the system, I support users examining each local pattern and comparing it with the global one both statistically and visually.

5.3 System Introduction

In this section, we introduce the proposed local pattern exploration method and our system design. In our system, we provide 5 different coordinated views to assist users in exploring the local patterns.

5.3.1 Global Space Exploration

The *global view* is designed to give users a global sense of the whole dataset. Basically, any multivariate data visualization techniques, such as scatterplot matrices, parallel coordinates, pixel oriented techniques, or glyphs, can be used to display and explore the data globally. Of these methods, only glyphs show each data point individually as an entity. We use a star glyph because the analyst can easily specify which individual data point he/she wants to examine, thus leading to a easy exploration of the local pattern of that data point. A major drawback for the glyph display method is the scalability issue. When the data size is very large, each glyph is very small and it is difficult to recognize and specify a single data item. A solution is to use brushing and filtering techniques to

hide uninteresting local patterns to save the display space. Another solution is clustering similar local patterns and displaying different clusters in separate views.

To assist analysts in discovering anomalous local patterns, i.e., a subgroup of neighbor data points that are different from the global pattern, we encode the statistical results using color. As shown in Fig. 5.2, gray color means there is no significant difference between the sample and the population in a significance level (p-value is larger than 0.05), suggesting the local pattern is not an anomaly. Red and blue colors mean that a significant difference is detected (p-value is less than 0.05). Red means the z-score is less than zero (the critical value is -1.96 for 0.05 level), which means the local pattern has significantly lower mean value than that of the global pattern. Similarly, blue means the z-score is larger than zero (the critical value is 1.96 for 0.05 level), indicating a higher mean value compared to the global pattern. We use a diverging color strategy for two colors from Color Brewer [16]; this strategy is also used in the local pattern view for comparative neighber representation. The darker the red and blue colors are, the higher the significance is (i.e., a smaller p-value is obtained). When users examine each individual local pattern, red and blue items are generally of users' interests. Though we use 0.05 as default significant level, if users only want to focus on the data items that are extremely different from the global pattern, they can change the significant level to a smaller value, such as 0.01 or 0.001, to reduce the number of anomalous local patterns, i.e., red and blue star glyphs.

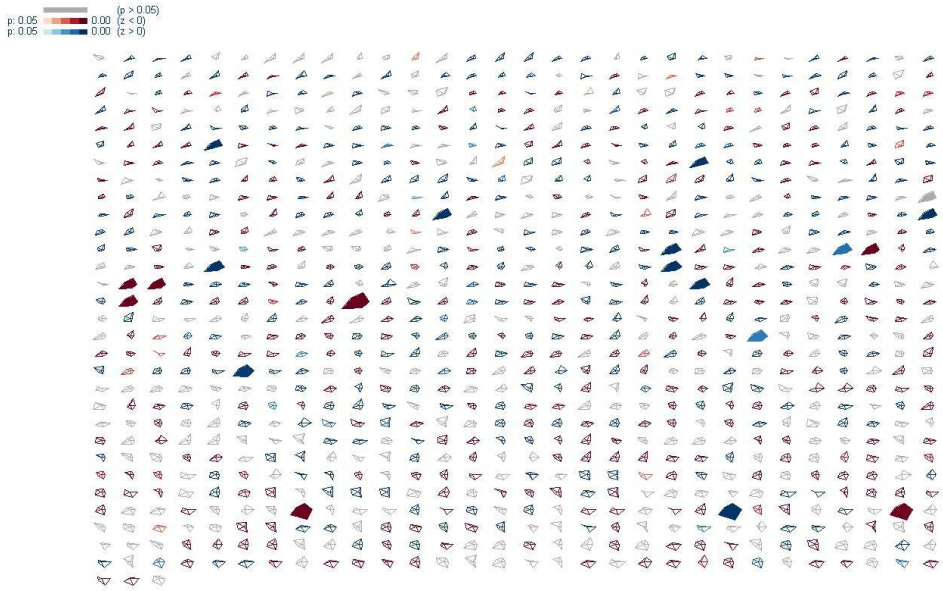

Figure 5.2: The global display using star glyphs (902 records from the diamond dataset). The color represents whether the data item is an anomalous local pattern or not. The filled star glyphs are selected local pattern neighbors.

When the user moves the cursor onto a single data item, its neighbors and the item

itself are highlighted using larger filled glyphs to draw the user's attention. Meanwhile, the basic statistical information is shown in the bottom bar, such as neighbor count, mean value, z-score, and p-value.

5.3.2 Local Pattern Examination

During the interactive exploration in the global view, when the user moves the cursor onto a data item, another view displaying all its neighbors and the selected point are drawn, called the *local pattern view*. The main purpose for this view is to illustrate the relationships between the focal point and all its neighbors. As a general solution, assume that the focal point is placed in the center of this view; all the neighbors' positions should be designated to reflect their relationships, according to different types of extracted local patterns and the users' requirements.

In particular, in our system, the focal point is shown in the center of the display using a star glyph, which allows the user to easily recognize the connection between the local pattern view and the global view. The two cross lines (vertical and horizontal) passing the center create four quadrants, using the focal point as the origin. As a layout strategy, we map the difference in target values between a neighbor and the focal point as Y, meaning for each neighbor, if its target value is higher than the focal point's target value, it is located in the upper half. Contrariwise, if the target value is lower than the focal point, it is located in the lower half. The higher the absolute difference is, the further away the neighbor is placed. This layout strategy tells users where to find an interesting neighbor when the goal is to discover a neighbor with different target attribute values, such as looking for a more/less expensive apartment.

As discussed before, the local pattern in this chapter is the orientation angle θ. The angle is mapped to X in this view. The angle of the focal point is $\pi/2$, assuming the direction conforms with the local trend. When the angle between a connecting vector and the normal vector of the local trend is less than $\pi/2$, the corresponding neighbor is placed in the left half of the view. If θ is smaller (larger) than $\pi/2$ it means the neighbor's target value is smaller (larger) than the estimate. The user can use this piece of information to discover interesting neighbors. For instance, taking the example of the apartment finding problem, given a focal apartment, the students should have more interest in the neighbor apartments shown on the left side, as those neighbors are detected by our system as having lower prices than predicted comparing with the focal point.

For each neighbor, we support two display methods. The first one is the original value display, which means that for each neighbor, the attribute values in the original dataset are shown. In this case, we again use the star glyphs to represent each neighbor, so that users can connect this view with the global view (Fig. 5.4). The second display method is a comparative display (Fig. 5.5), in which the focal point is the base line, represented as m dashes, where m is the number of attributes. For each neighbor, there are m bars corresponding to its m attributes, where a upward (downward) bar for an attribute indicates that the neighbor's value in that dimension is higher (lower) than that of the focal point. This piece of information is also redundantly represented using colors: blue

means higher and red means lower. The larger the difference is, the darker the color is. Note that the height of a bar represents the difference between the neighbor and the focal point in the normalized space, so that when the relationship between the neighbor and the focal point conforms with the local trend, the sum of the bar heights of the independent attributes are the same as the bar height of the target for that neighbor.

In terms of the scalability for the comparative display, both the number of neighbors and the number of dimensions can result in visual clutter and overlapping. For example, Figure 5.3 shows the local pattern view with a large number of neighbors (332 neighbors in total). One simple solution for a fair number of neighbors is to allow users to interactively change a scale factor to reduce the size of each data item. A data item will be enlarged to its original size when the user moves the cursor onto it. Another solution is to reduce the number of the displayed neighbors: the users could specify a small number of k, and only the most closet k neighbors or the most interesting k neighbors, based on a certain interestingness function, are displayed in this view. We can also apply a clustering technique to reduce the number of displayed neighbors. That means we can cluster to group nearby similar neighbors. After that, each displayed neighbor is a visual representation of a set of similar neighbors. Some interactions can be integrated allowing further exploration of a specified group of similar neighbors, or allowing the user to adjust the level of cluster tree for displaying. When there is a large number of attributes, a dimension reduction or selection technique could be applied before analysis. For example, the attributes with lower influencing factors can be removed for reducing visual clutter.

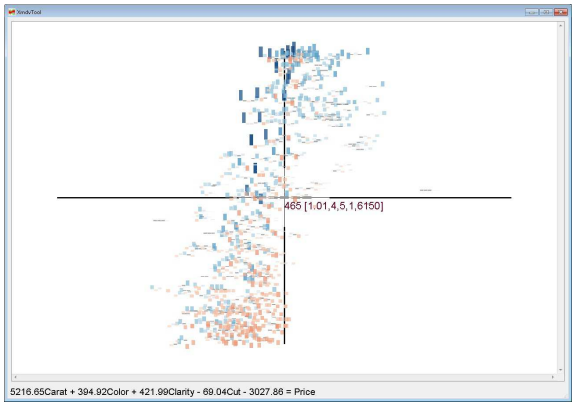

Figure 5.3: The local pattern view with a large number of neighbors (332 neighbors), which results in visual clutter.

The local regression line in an equation form is shown in the bottom bar to assist the analyst in performing sensitivity analysis. For the interactions in this view, when the user moves the cursor on the focal point or one of the neighbors, the data item ID and its attribute values are displayed next to it (in the form of ID[attribute 1, attribute 2, ..., attribute n]). The user can click any data point to show or hide the attribute values.

Figure 5.4: Neighbor representation using original values.

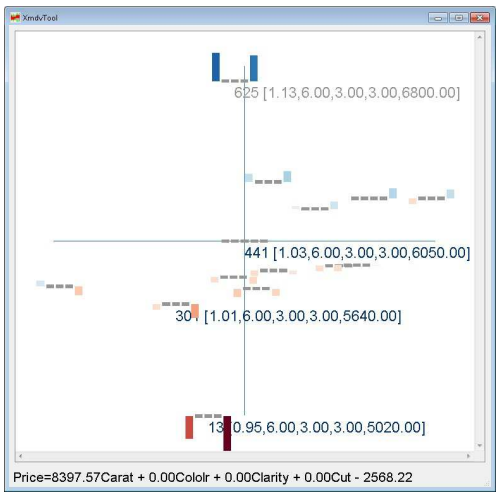

Figure 5.5: Neighbor representation using comparative values.

5.3.3 Compare the Local Pattern with the Global Pattern

The colors of each data point in the global view represents the statistical test results, i.e., an outlier factor indicates how likely the local subgroup is different from others. However, knowing the statistical test results is often insufficient. For example, some insignificant results may also be interesting due to a large deviation. Therefore, a visual comparison of the local with the global is still needed. To allow the user to compare the local pattern with the global pattern both statistically and visually, we provide users a *comparison view*, showing the global distribution (directions to neighbors) using a histogram. The mean values and confidence intervals for both the global and local pattern are also shown in the bottom (Figure 5.6). The use of this view is shown in the case study section.

5.3.4 Adjusting the Local Pattern

The local partial derivative values reflect how the independent variables influence the target variable in the local area. However, the derivative values may not necessarily meet the user's expectations and requirements when they want to find interesting neighbors. For instance, assume that the students wants to move to another apartment from the current one and are willing to increase their payments, e.g., they would be willing to pay around $100 more for one more bedroom, or pay $100 more for moving a mile closer to the campus. In this case, one more bedroom is the same as 1 mile closer, in terms of influencing ability on the target. For different users, the requirements are likely different. Students with cars may prefer larger apartment, while ones without cars prefer a closer apartment. In the first case they would like to increase the influencing factor of size on price, while in the second case, they would like to increase the influencing factor of distance. It means that different users have different ways to define "better" when they want to find "better"

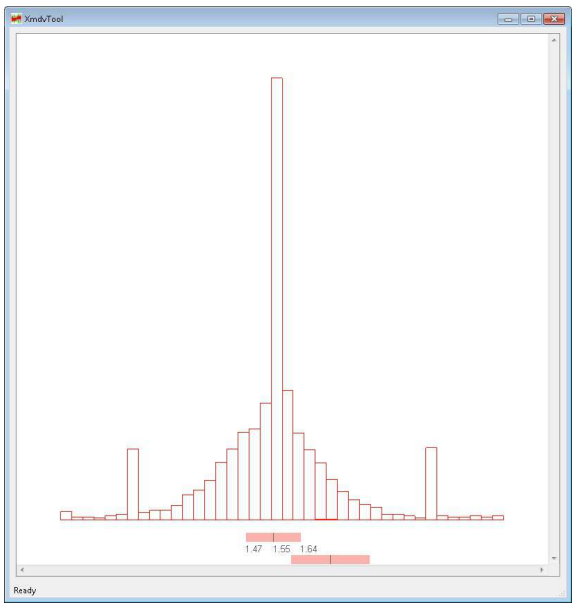

Figure 5.6: The comparison view. The two pink bars in the bottom represent the confidence interval of global pattern (upper) and selected local pattern (lower).

neighbors around the focal point.

In our system, we provide users a *local pattern adjusting view*, using parallel coordinates (Fig. 5.7). The partial derivatives are drawn as a poly-line. The user can interactively change the coefficient values, i.e., the slopes of the trend line, by dragging the poly-line on each axis. During the adjustment, the local pattern view is also dynamically changed to reflect the new relationships among the focal point and its neighbors in the new "environment", i.e., using the new trend. This is because we calculate the relationships among the focal point and its neighbors based on the normal vector of the hyperplane. Since we define the standard direction using the normal vector, we can understand this tuning as equivalent to changing the definition of north in a map.

Figures 5.8 and 5.9 show the local pattern view before and after changing the coefficients. The dataset is a car sales dataset (from the SPSS sample datasets). For easier understanding, only two independent attributes are considered: horsepower and MPG. The target is the price of the car. The goal is to compare a neighbor car, whose ID is 68 (the upper one with attribute values) with the focal one (ID is 72). It is shown that locally horsepower influences the price positively. Before adjusting, this neighbor is in the right hand side, which means a worse deal since the price is higher than estimated. We can recognize this by the comparative display of the neighbor; the sum of the height of the independent attribute bars is less than the target bar height (a lower bar for horsepower than the bar for price), which means the price is higher than estimated. After changing the weight (coefficient) of horsepower to a higher value, this neighbor become a better deal (in the left side). This is because the customer considers horsepower as an important attribute. After changing, the sum of the bar heights for independent attributes increases

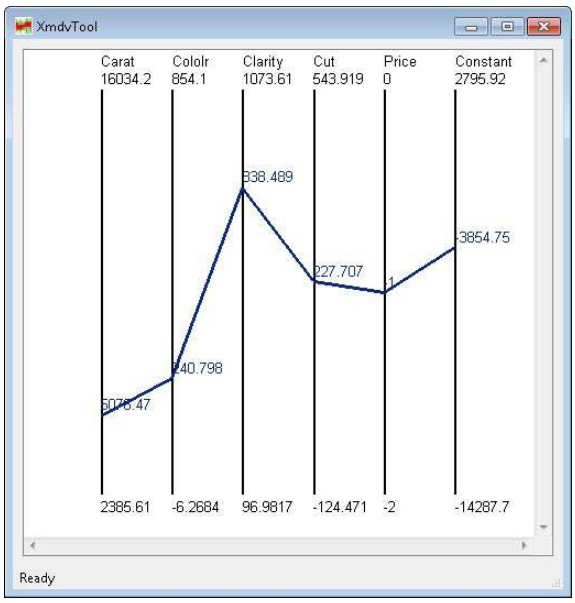

Figure 5.7: The local pattern adjusting view. The poly-line represents the adjustable coefficients.

and exceeds the target bar height. This example shows users can customize and change the coefficients according to their priorities.

5.3.5 Integrate the Local Pattern into the Global Space View

Generally, a local pattern is a value or a vector for a single focal point. Thus, the local pattern vector can be treated the same as the attribute values in the original data space. Assume there are n independent attributes and 1 target attribute, we can create n new dimensions taking the derivative values as derived dimensions and integrate them into the original attributes, thus resulting in a new dataset with $2n + 1$ dimensions. Any multivariate data visualization technique can be used to display this new dataset, such as scatterplot matricies and parallel coordinates. This visualization enables users to discover the relationships among the derived dimensions and the original dimensions.

Fig. 5.10 shows an example of the *integrated view*. In this example, each data point is a child's basic body information: age, gender, height and weight. The age range is between 5 and 11. We use weight as the target and the goal is to discover for children of different ages and genders, how height influences weight. The neighbors are defined as children with the same age and gender, and similar height and weight. The figure shows the distribution of the derivatives ($\partial weight/\partial height$) in the original space (age and gender). The derivative values are color-coded (darker color means higher value) and the points are jittered to avoid overlaps. We can discover that the derivatives increase as age increases. Analysts can also compare the derivatives for different genders to answer questions, such as for 8-years-old children, which gender has larger derivative values (the

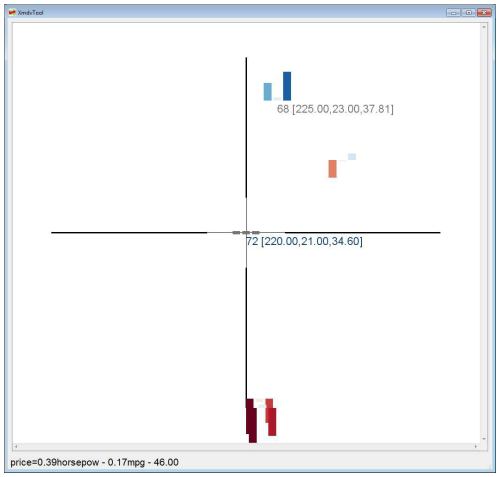

Figure 5.8: The local pattern view before adjusting the horsepower coefficient. The neighbor (ID 68) is a worse deal.

Figure 5.9: The local pattern view after adjusting the horsepower coefficient. The neighbor (ID 68) became a better deal.

answer is female).

5.4 Case Study

In this section, we discuss case studies to evaluate our approach and show the effectiveness of our system. The dataset is a diamond dataset obtained from an online jewelry store [41]. Each data item is one diamond. The target attribute is *price*. There are 4 different independent attributes that influence the price of a diamond: *weight* (*carat*), *color*, *clarity* and *cut*. The goal is to assist customers in choosing a diamond. The discovery can also tell the retailer whether the price of a certain diamond is set appropriately. We use a subset of the diamonds with a certain price range ($5000 - $8000), since we assume that customers have a budget range for shopping, rather than caring about the whole dataset. The whole dataset has 13298 data items and the subset has 903 data items.

The main computational bottleneck is in the calculations involved in finding neighbors, which would be performed in a $O(n^2)$ time cost without any index data structure, assuming the data size is n. After the neighbors for each data item are found, the least square linear regression cost is $O(Km^2)$, where K is the average neighbor count and m is the dimension number. During the exploration of each local pattern, there is no computational cost since the neighbor index is already created. Another cost in our system is in the local pattern adjusting period, which is $O(k)$ (k is the neighbor count of the examined focal point). On a 3 Ghz dual core desktop PC with 4 GB of RAM and an ATI Radeon X1550 graphics card, we ran our system both for the whole dataset and the subset of the diamond dataset (neighbor range is defined as 0.1 of the entire range of each attribute). For the subset, the time for finding neighbors and regression calculating took less than 2

Figure 5.10: The view for integrating derivatives into global space. The jittered points with different colors indicate the coefficient of $\partial height/\partial weight$. As age increases, the coefficient increases. For the same age, the coefficient values are different for different genders.

seconds. For the whole dataset, the time required is about 6 minutes. The huge difference is mainly due to the quadratic time complexity for finding neighbors. For both datasets, the exploration of local patterns, as well as local pattern adjustment, can be performed and updated in real time.

5.4.1 Where are the Good Deals

For easier understanding, we start from a single independent attribute *weight*. The user of our system can achieve this by defining an appropriate neighborhood: two diamonds are neighbors when they have similar *weight* and *price*, as well as they are of the same *color*, *clarity* and *cut*. The extracted local pattern is the orientations to the neighbors. Fig. 5.2 shows the global star glyph display. The color indicates whether the diamond is an anomalous one. To examine the global distribution, the user can open the comparison view (Fig. 5.6). The global distribution is similar with a normal distribution, except there are two peaks on each side. We will show later this is due to some anomalies, i.e., some diamonds whose prices are not set appropriately. The mean of the distribution is about $\pi/2$, which is the same as we discussed before, assuming the neighbors fit the local linear trend .

To understand the normal and abnormal data items in detail, we show three local pattern views for gray, red, and blue data points. Figure 5.11 shows the local pattern view of a gray data point. All the neighors of this data point are in the center of the view (x

position), indicating that the directions to the neighbors are all about $\pi/2$. This means that all the data points in the local area fit the regression hyperplane, which is very common in the dataset. We can also recognize this local fitting by the comparative representation of all neighbors: the height of the first bar (*weight*) is almost the same as the height of the last bar (*price*). This indicates the price difference, between the focal point and one neighbor, is proportional to the weight difference. To assist the analyst in performing the sensitivity analysis, i.e., what is the change rate of the target as an independent attribute value varies, we show the local regression model in the bottom bar. It is shown that in this local area, as the weight increases, the price increases, which means a positive influencing factor. The changing rate of price is \$55, as the weight increases 0.01 carat. The influencing factors of the other independent attributes are all 0, since all neighbors have the same values.

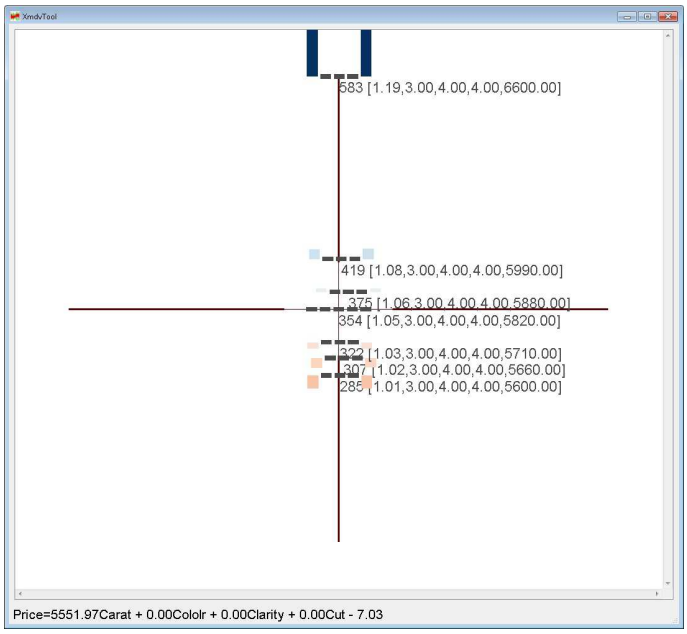

Figure 5.11: The local pattern view of a gray data item. The orientation from the focal point to all its neighbors are $\pi/2$, which is common in the dataset.

Figure 5.12 shows the local pattern view for a diamond that is blue in Figure 5.2, suggesting that it is an anomaly and the test result shows the mean of this local pattern is significantly higher than the global pattern. The user can see that all the neighbors are in the right half of the view. This means that for each neighbor, the direction is larger than $\pi/2$. From the discussion before, we know that when the direction is larger than $\pi/2$ for a certain neighbor, it means the target variable is higher than estimated, assuming the local regression plane passes through the focal point. In particular, the local sensitivity shows that as weight increases 0.01 carat, the price increases \$118. However, the price of the local neighbors are higher than estimated considering this changing trend. Take the upper diamond for example. The upper half means a higher target value based on our local pattern layout strategy. We can see that for this neighbor, the weight is 0.01 carat

higher than the focal point, while the price is $450 higher than the focal point, which is a larger changing rate, compared with the local trend. The user can also read this from the comparative representation of this neighbor: a higher and darker bar for price than the bar for weight, which means the price change rate is higher than weight. This tells users that this neighbor is a worse deal compared with the focal point. Similarly, we can consider another neighbor whose price is lower than the focal point, i.e., in the bottom half of the display (the nearest one to the focal point). The neighbor's weight is 0.02 lower than the focal point. If this neighbor fits the local trend, the price would be $118*2=$236 lower than the focal diamond. However, the price is only $120 lower than the focal diamond, which also means this neighbor is not a good deal compared with the focal diamond. The user can also read this through the comparative representation of this neighbor: a much darker and lower bar for weight than the bar for price. From these discussions, we know that for blue diamonds, generally most of neighbors are in the right half side of the view, which means there are worse deal compared with this one. Thus, the blue diamonds should be preferable for the customers.

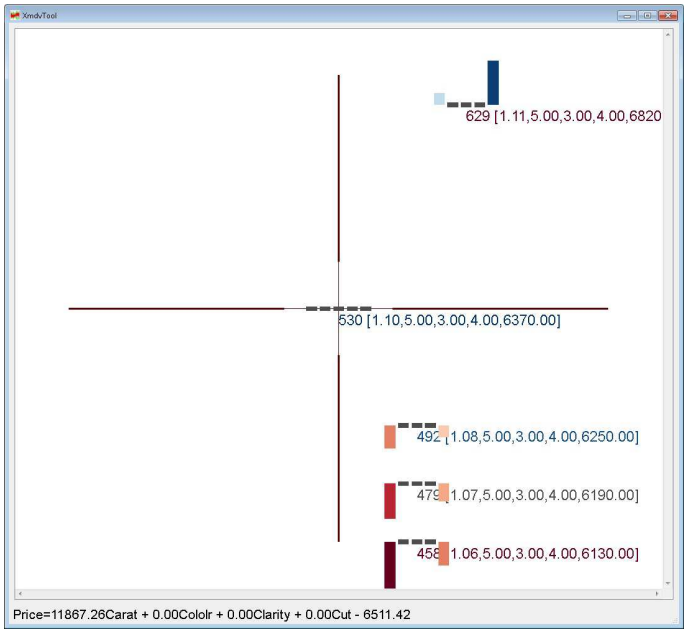

Figure 5.12: The local pattern view of a blue data item. The orientations from the focal point to most of its neighbors are larger than $\pi/2$, which means the neighbors' target values are higher than estimated. In other words, the focal point is a "good deal".

Finally, we give an example of a diamond mapped to red in Fig. 5.2. Similar with the discussion for blue diamonds, a red diamond means there are many better deals compared with this one. Fig. 5.13 shows the local pattern view of a red diamond. It is shown that locally as the weight increases 0.01 carat, the price increases $332. The two neighbors (with attribute values) are better than this one. For the upper neighbor, the weight is the same as the focal point, while the price is $570 lower than the focal point (a downward

red bar). For the lower neighbor, the weight is higher than the focal point, while the price is $150 lower than the focal diamond. For the focal diamond, the neighbors in the left half are better recommendations. Since there are many blue and red diamonds (anomalies), the distribution of the global pattern has two peaks in each side. From the retailer side, it should consider decreasing the prices of the red diamonds and increasing the prices of blue diamonds.

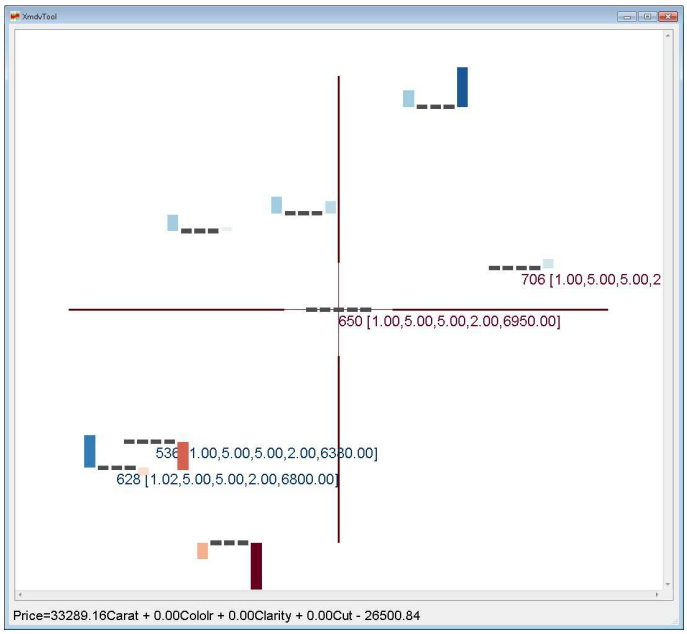

Figure 5.13: The local pattern view of a red data item. The orientations from the focal point to most of its neighbors are lower than $\pi/2$, which means the neighbors' target values are lower than estimated. In other words, the focal point is a "bad deal".

This method of discovering good and bad deals in this dataset is also suitable for more than one independent attribute. We choose only one independent attribute just because it is easy to verify whether the diamonds are worth buying.

5.4.2 Display the Local Pattern in the Global View

It is shown that for different local patterns (subsets of neighbors), the price increases differently as the weight increases. This means the coefficients ($\partial price/\partial weight$) are different in the whole space. It is useful to give users a global sense in terms of how the sensitivity derivatives are distributed in the original space. To assist users in better understanding this, we use the whole dataset rather than a subset of a certain range. Fig. 5.14 shows a scatterplot view of the dataset. We use color to represent the derivative values: dark blue means high and dark orange means low. The color strategy is again diverging. The points are jittered to reduce overlapping.

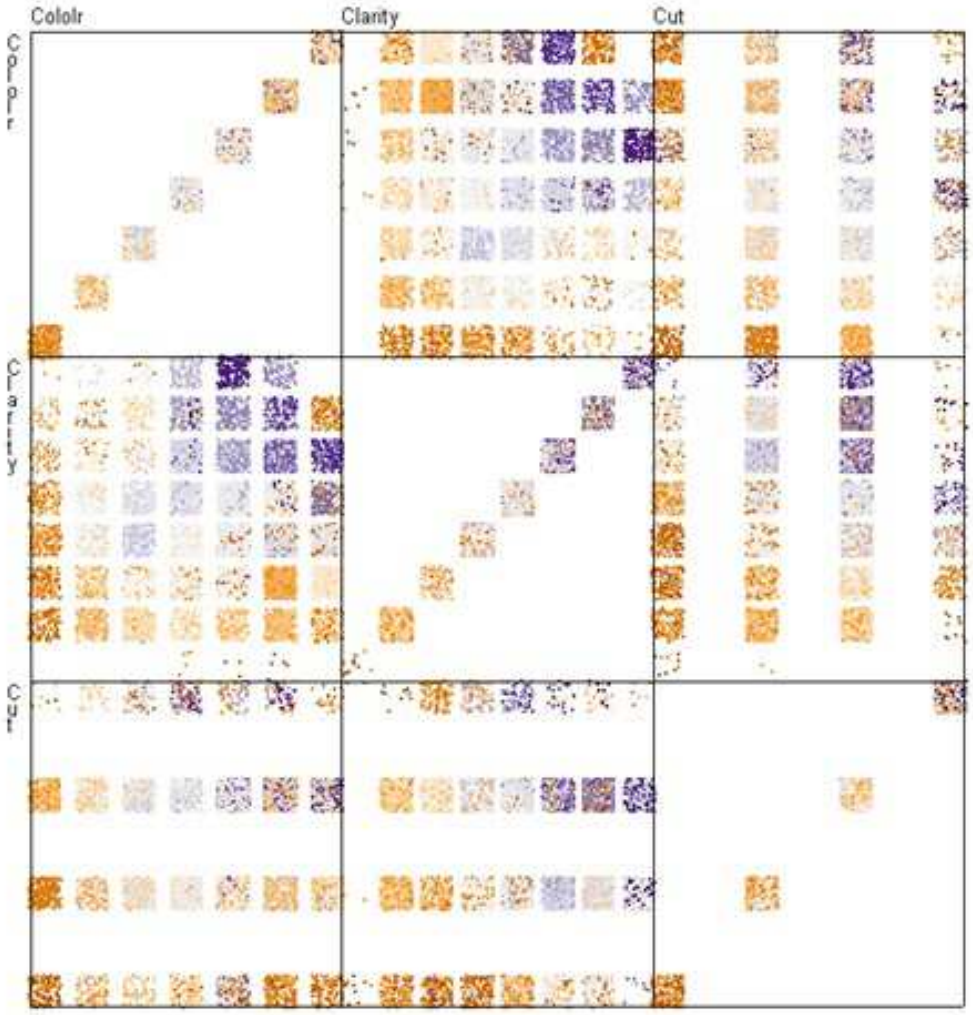

Figure 5.14: The coefficients of $\partial price/\partial weight$ are color-mapped and displayed in a scatterplot matrix of original attribute space.

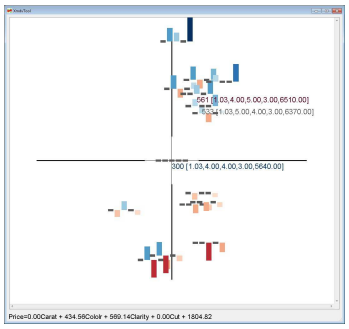

Figure 5.15: The local pattern view before tuning the coefficients. One neighbor (ID 533) has higher *color* and the other neighbor (ID 561) has higher *clarity*.

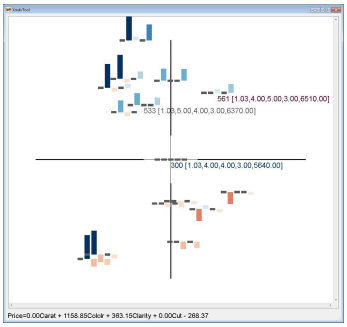

Figure 5.16: The local pattern view after increasing the coefficient of *color* and decreasing the coefficient of *clarity*. The neighbor with higher *color* became a "good" deal.

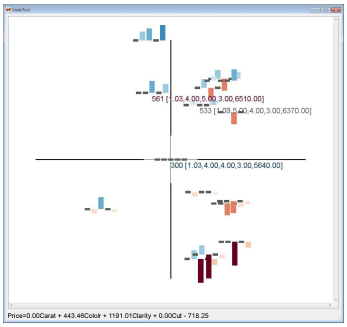

Figure 5.17: The local pattern view after decreasing the coefficient of *color* and increasing the coefficient of *clarity*. The neighbor with higher *clarity* became a "good" deal.

Users can discover that the derivatives are pretty consistent for diamonds of the same color, clarity and cut. This means that for different subset of neighbors, although their weights and prices are of different ranges, the influencing factors of weight on price are very similar. Another discovery is that as color, clarity and cut increase, the derivatives generally increase (from dark orange to dark blue). This means that for diamonds of higher quality, the weight is more important for price, i.e., the price is very sensitive with changing weight for the subspace of higher color, clarity and cut. As customers, when they notice that, they could consider changing their choices based on this discovery. For the blue region, they can consider choosing a diamond of lower weight, since it will save them a lot of money. In contrast, for the orange region, they can consider choosing a diamond of higher weight, since it won't increase their costs too much. We can also notice that in the upper right of the plot of clarity vs. color, there is a dark orange block in the blue area. A possible explanation for this divergence from the main pattern is that there are not enough diamonds in this region, whose color and clarity values are both very high. The low price variance results in low coefficient values.

5.4.3 Customize the Local Pattern

Given budget limits, customers have to find a trade-off when considering the diamond attributes. Although the extracted sensitivity coefficients reflect locally how the price is influenced by the diamond attributes, when customers are selecting a diamond, they have their own attribute priorities. It means that they have to sacrifice the unimportant attributes (decrease the values) to reach the higher configurations on their preferred attributes. For example, some customers may prefer higher weight diamonds (larger ones), while not caring too much about the clarity; and some may prefer higher cut (more shininess),

while not caring too much about the color. In different cases, customers have their own ways to define the meaning of "good". Thus the customers should be able to customize the model (sensitivity coefficients) and find good diamonds in different cases.

We show an example to illustrate how customers can customize their requirements. Assume that a customer has decided the weight and cut of the selection, and is struggling with higher color or higher clarity. In this case, the neighborhood is defined as diamonds of the same weight and cut. For color and clarity, the neighborhood region covers three levels of each, indicating lower, current, and higher values. Fig. 5.15 shows the local pattern view of a preferable diamond before adjusting the coefficients. The two neighbors, shown with attribute values, are two alternative options compared with the focal one. Both of them are more expensive than the focal one: one has higher (better) color and one has higher (better) clarity. Before tuning the coefficients, none of them are better deals (in the left half). If the customer knows that she prefers higher color (clarity), she can accordingly increase the coefficient for color (clarity) and/or decrease that for clarity (color). Fig. 5.16 and Fig. 5.17 show the local pattern views after adjusting the coefficients. In Fig. 5.16, the coefficient for color is increased and the coefficient for clarity is decreased. It is clear that the neighbor with high color became a good deal. These two neighbors can be easily differentiated and the customer can tell which one is worth purchasing in this circumstance. A similar result is shown in Fig. 5.16. In this case, the coefficient for clarity is increased and the coefficient for color is decreased. We can discover that the two neighbors shift in the opposite directions compared with Fig. 5.16. According to this example, we can see that customers can define "good" when selecting a diamond. Generally speaking, for any other type of local patterns, users can customize the definition of "interestingness" and the system is able to provide users different recommendations of neighbors.

5.5 User Study

In this section, we discuss a user study for evaluating the effectiveness of the visual representations of the local pattern. We focused on two visual factors: different types of glyph representations and different layout strategies. To remove the interaction effects among the two factors, we evaluate the two factors independently.

For the glyph type, our goal was to examine the effectiveness of the comparative display, i.e., using upward and downward bars to represent the relationship between the focal point and its neighbors. To compare with other methods, we implemented two other types of commonly used glyph representations: profile glyphs (Figure 5.18) and star glyphs (Figure 5.19). To make the comparison fair, we also integrated the same color strategy into these two glyph types. Our hypothesis was that the comparative glyph method better reveals the relationships between the selected focal point and its neighbors. A sample question was "Compared to the focal diamond, how many neighbors have both lower color and lower clarity?"

For the layout strategy, our goal was to examine the effectiveness of the local pattern

Figure 5.18: The profile glyph display.

Figure 5.19: The star glyph display.

view layout, namely, placing the selected focal point in the center and placing the neighbors in the four quadrants according to the interestingness (such as diamond price). For comparison, we implemented a scatterplot display that maps the attribute values to the x and y locations. The focal point was differentiated by both size and color. Our hypothesis was that the centered layout can better help analysts locate interesting neighbors. A sample question was "How many more dollars are needed to buy a diamond with both higher color and higher clarity?" The dataset we used is the same as the dataset mentioned in the case study which had 4 independent attributes and 1 target.

We invited students to be the subjects (21 in total) in the user study. The subjects were asked to answer 8 questions about local patterns based on visual representations. In this user study, we didn't ask the subjects to use our system because the main goal was to evaluate the local pattern design method. In Section 5.6, we describe in detail how a user explored a dataset using our system. The subjects answered the questions based on screen-copied figures printed out on paper. Note that any single question could be answered based on different visual representation methods of the same local pattern, such as different glyph types or different layout strategies. Subjects were randomly assigned a visual representation method to answer a given question. Take the evaluation of the layout strategy for example. We designed two questions (question Q_a and question Q_b) to compare the two layout methods. We generated two groups of questions, group G_A and group G_B, as follows. Each question group had both questions Q_a and Q_b. In group G_A, question Q_a would be answered based on the designed local pattern layout strategy, while question Q_b would be answered based on the scatterplot layout. In group G_B, the questions are the same, but we exchanged the layout strategies: question Q_a was represented using the scatterplot and question Q_b was represented using our local pattern layout method. In the study, we randomly assigned half of the subjects to question group G_A and the other half to question group G_B. Similarly, we generated three groups of questions to evaluate the glyph types because there are three different glyph representations.

Before the study, the subjects signed a consent form. Then each subject was shown a brief explanation of the study using examples and sample questions, such as which dataset we used and how to read the figures. The subjects finished the study by answering several questions. We recorded the time each subject spent on each question for further analysis.

Figure 5.20 uses error bars with a 0.95 confidence interval to show the accuracy for the three glyph types. We found that the comparative glyph and the profile glyph were very similar in terms of accuracy. It is clear that both the comparative glyph and the profile glyph are much better than the star glyph: the p-values are 0.017 and 0.023, respectively.

We also examined the time spent for each glyph type and the result are shown in Figure 5.21. Similarly, the comparative glyph and profile glyph are better than the star glyph. The difference between comparative glyph and star glyph is significant (p-value=0.026). Although there is no significant difference between comparative and profile glyphs (p-value=0.232), the time subjects spent on the comparative glyph was much lower than for the profile glyph. To conclude, we found comparative glyphs and profile glyphs were better than the star glyphs for both accuracy and time. The accuracy for comparative glyphs and profile glyphs are very similar, but they spent more time on profile glyphs.

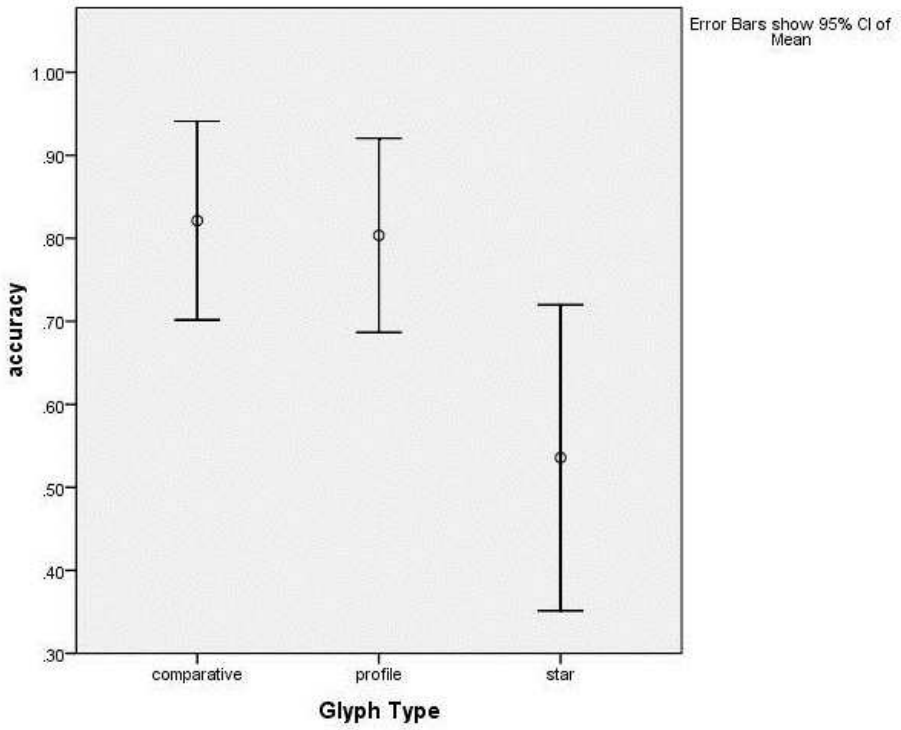

Figure 5.20: The comparison of accuracy for different glyph types.

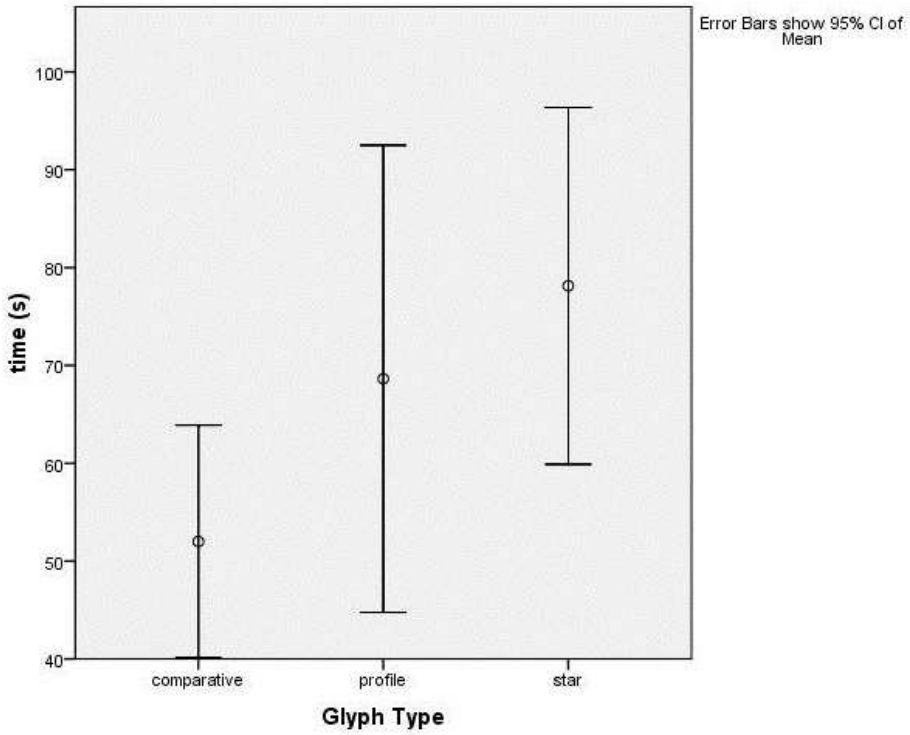

Figure 5.21: The comparison of time for different glyph types.

Lastly, we compared the two layout strategies. Figure 5.22 compares the accuracy for these two layout methods. In terms of accuracy, the two strategies are almost the same (nearly 80%). However, in terms of task completion time, we noticed that the subjects spent much more time when using the scatterplot layout. The average time for the centered layout was 62 seconds, while for the scatterplot layout it was 87 seconds, which is shown in Figure 5.23. This is a statistically significant difference (p-value=0.038). We also noticed that the time variance of the centered layout is large. We believe this is because of different learning rates for this new layout method. Some subjects seemed to learn and get used to this layout very quickly, while others had difficulties and spent more time getting used to it. In a future evaluation, we will try to confirm this difference in learning rates and repeat the tests with trained subjects.

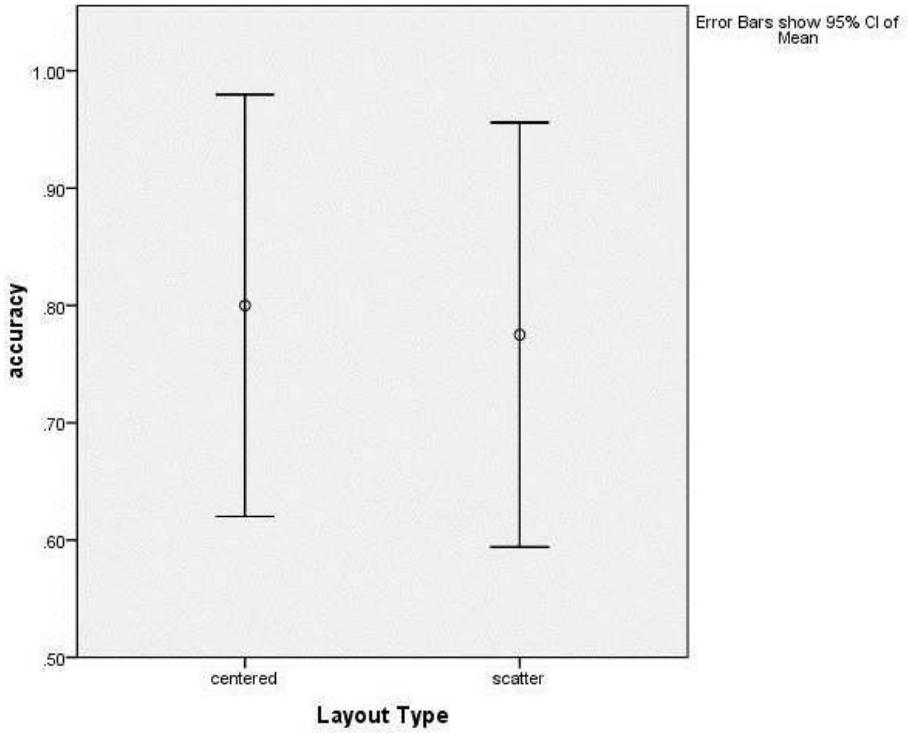

Figure 5.22: The comparison of accuracy for different layout types.

5.6 Usage Session

We now demonstrate how our visual exploration method could be used for solving real life problems. Our usage session was again based on the diamond dataset. We invited a user who was trying to make a decision on buying a diamond to test our system.

Before using our system, he first browsed some on-line diamond selling websites on the internet to get familiar with the diamond purchasing task. There were two reasons

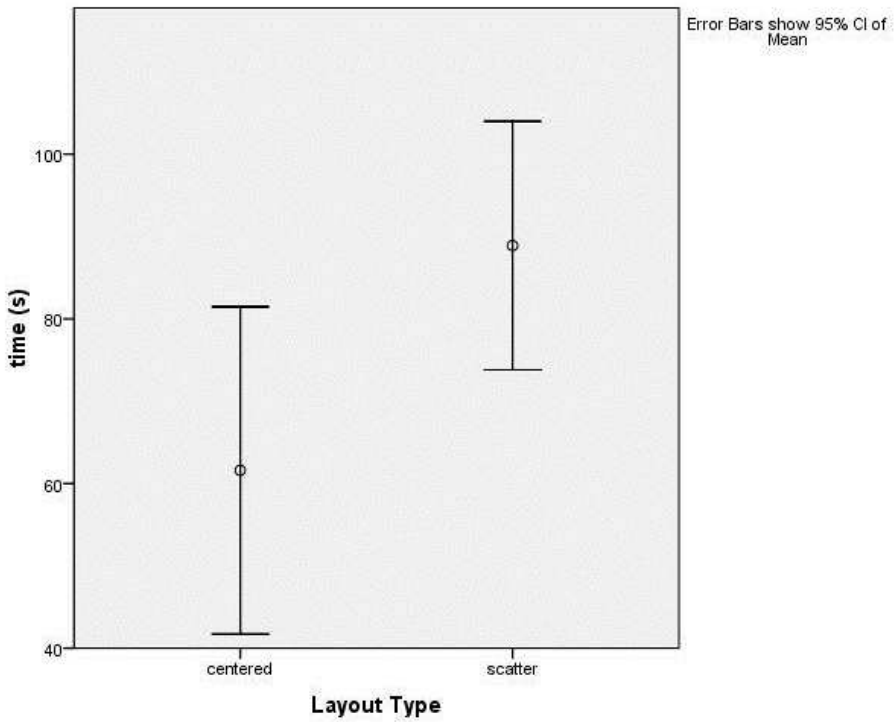

Figure 5.23: The comparison of time for different layout types.

for this activity prior to using our system. The first was to help him understand which attributes are important to him, i.e., to develop a personal preference. The second reason was that he could determine the minimum requirements and the price range he'd like to choose from. He told us his preferred price range was roughly between $6000 and $7000. In terms of the importance of different attributes, he thought weight (size) was the most important one. The second attribute important to him was color. The other two attributes, clarity and cut, were not very important to him. He said this was because he thought the latter two attributes were not as noticeable as weight and color for him. He also indicated minimum requirements on these attributes: weight needed to be at least 1.1; color needed to be at least H (the required value was 4 where the best color value is 8); clarity needed to be at least SI1 (the required value was 3 where the best clarity value is 8); he did not have any requirements on the attribute cut.

With these requirements and preferences, he started using our visual exploration system to perform the task. The first step was to define the local neighborhood range. After being given some explanation on this step, he decided to define two diamonds as neighbors when they have similar weight (within 0.15), color (plus or minus 1) and price (within $500). He did not care about the other two attributes, clarity and cut, so he decided to remove their influence at this step.

He then explored in the data in the global view (the star glyph display) by hovering the cursor over the glyphs (data items). The data attributes are shown when the cursor

is on that data item. The glyphs are ordered based on price, so he roughly picked some interesting candidates within his preferred price range. He had two criteria when choosing the candidates. For the first one, since he considered weight the most important attribute, he picked several heavy (large) diamonds. The second criterion was to focus more on the blue data items. This is because we told him that generally glyphs colored blue are usually better deals. After this initial rough selection, he chose three candidates as shown in Table 5.1. These three diamonds are all blue, i.e., they are better than most of their neighbors.

ID	Weight	Color	Clarity	Cut	Price
584	1.26	6	2	3	6600
567	1.52	6	1	3	6510
544	1.51	2	3	2	6420

Table 5.1: Candidate diamonds after a rough exploration in the global star glyph view.

Then he decided to refine his selection by examining each candidate in the local pattern view. He opened the local pattern view and compared the pre-selected candidates with their similar local neighbors. The three local pattern views of these candidates are shown in Figures 5.24, 5.25, and 5.26. The attributes are in the same order as introduced in section 5.4: the first attribute is *weight* and the last attribute is *price*. He wanted to find more interesting candidates on the left hand side in this view.

When he viewed the local neighbors of diamond 584, he noticed that diamond 624 was also a good choice because its weight is higher. Although the price is a little higher, since it is on the left hand side, it may still be worth buying. The second neighbor he was interested in was diamond 547. This diamond has the same weight as diamond 624, but it is much cheaper. Another interesting neighbor was diamond 461, whose weight is higher than candidate 584, but much cheaper. All three interesting neighbors are on the left hand side, indicating they are worth buying compared to the candidate diamond 584. Therefore, at this point, he removed diamond 584 from the candidate list and added the three newly found diamonds onto the list.

Then he opened the local pattern view for diamond 567. He noticed that the neighbor diamond 400 was a much better choice. The weight and color are both better than those of the previous chosen diamond 567, yet with a lower price. So he removed 567 from the candidate list and added diamond 400 to it. He didn't find any interesting neighbors for candidate diamond 544.

Next, he wanted to view the local patterns of the newly added candidates to further enlarge the candidate list with more choices. He didn't find any better choices for diamonds 624, 547, and 400. When he viewed the local pattern of diamond 461 (Figure 5.27), he found an interesting neighbor, diamond 384. Because its weight is higher and its price is lower, he added it to the list. The candidate list at this point is shown in Table 5.2. Notice that after refinement, he had found several additional interesting candidates and only one pre-chosen diamond survived after examining the neighbors.

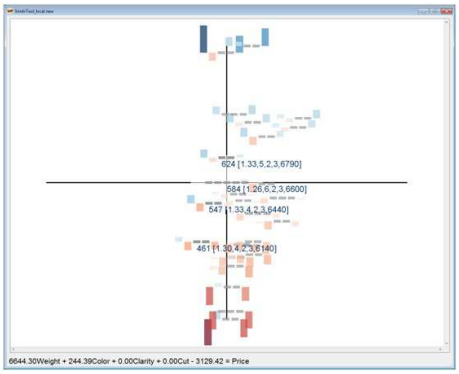

Figure 5.24: The local pattern view of diamond 584.

Figure 5.25: The local pattern view of diamond 567.

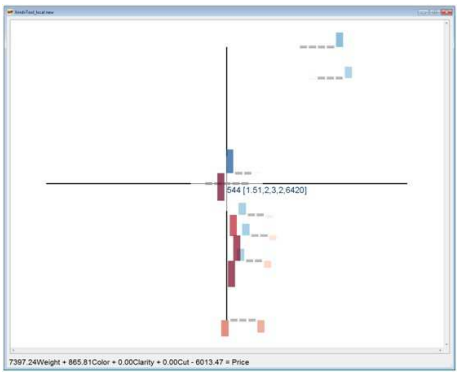

Figure 5.26: The local pattern view of diamond 544.

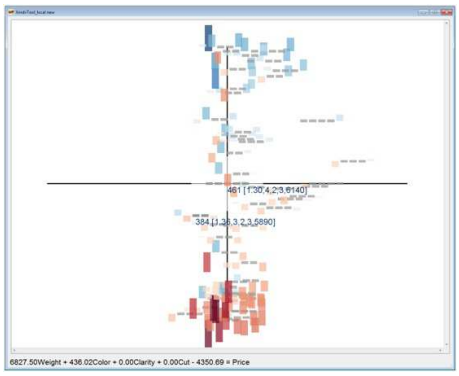

Figure 5.27: The local pattern view of diamond 461.

ID	Weight	Color	Clarity	Cut	Price
384	1.36	3	2	3	5890
461	1.3	4	2	3	6140
624	1.33	5	2	3	6790
400	1.58	7	1	3	5940
544	1.51	2	3	2	6420
547	1.33	4	2	3	6440

Table 5.2: Candidate diamonds after examining each local pattern of the pre-selected diamonds.

He then made a final decision among candidates on this list. He first removed diamonds 544 and 384 because their color, an important attribute, did not satisfy his minimum requirement. He then removed diamond 400 because its clarity was lower than that of the rest. After this, he noticed that all the candidates' clarity were lower than his initial requirement. Since he cared about weight and color much more, he decided to make a compromise on clarity, i.e., reduce the minimum requirement from SI1 (value 3) to SI2 (value 2). Now he narrowed his choices down to three similar diamonds: ID 461, 624, and 547. He decided to remove diamond 624 because the price was high compared to the other two. After a careful comparison between diamonds 461 and 547, he finally decided to purchase diamond 461. This is because diamond 461's weight is only slightly smaller than diamond 547, which is probably not noticeable, but the price is $300 cheaper.

After the study, he said that overall this system was very helpful. The local pattern view helped him compare similar data items, find more interesting candidates, and guide him to make a more comprehensive decision. He mentioned that the system was easy to use and helped him finish the task very quickly.

We asked him whether he had some suggestions for improving our system. He pointed out some limitations and gave us some useful suggestions. He said the neighbor definition in the parallel coordinate view is somewhat confusing and he had difficulty understanding it. He said sometimes given a candidate, he only wanted to examine the neighbors with higher weight or color. He suggested we could add a function so that the user can dynamically change the neighbor definition and give him greater flexibility in defining neighbors not only centered in the focal diamond, but also can take the focal diamond's value as maximum or minimum, such as only cheaper neighbors.

Another suggestion was a sorting functionality. He said he might want to sort the star glyphs in the global view during exploration. This functionality is not currently supported but would not be difficult to add. A filtering functionality was also mentioned. He told us that a range query filter would be useful. It could be used to hide the less interesting diamonds which don't satisfy the minimum requirement. This functionality could be effective, especially in the case when a large number of local neighbors exist. The last comment was to have a comparative view for the selected candidates. The view could provide him an overall comparison, where he could select any of the candidates as the focal diamond.

5.7 Conclusion

This chapter presented a novel pointwise visualization and exploration technique for visual multivariate analysis. Generally, any local pattern extracted using the neighborhood around a focal point can be explored in a pointwise manner using our system. In particular, we focus on model construction and sensitivity analysis, where each local pattern is extracted based on a regression model and the relationships between the focal point and its neighbors. Using this system, analysts are able to explore the sensitivity information at individual data points. The layout strategy of local patterns can reveal which neighbors

are of potential interest. Therefore, our system can be used as a recommendation system. During exploration, analysts can interactively change the local pattern, i.e., the derivative coefficients, to perform sensitivity analysis based on different requirements. Following the idea of subgroup mining, we employ a statistical method to assign each local pattern an outlier factor, so that users can quickly identify anomalous local patterns that deviate from the global pattern. Users can also compare the local pattern with the global pattern both visually and statistically. We integrated the local pattern into the original attribute space using color mapping and jittering to reveal the distribution of the partial derivatives. We discuss case studies with real datasets to investigate the effectiveness and usefulness of our approach. We performed comparative evaluations to confirm our glyph design and layout decisions, and described the experience of a user performing a real task with the system.

Chapter 6

Conclusions

6.1 Summary

In this dissertation, I discussed three different visualization systems that assist analysts in visually discovering interesting patterns in multivariate datasets. The main goal is to discover patterns both computationally and visually. The proposed systems can facilitate retrieving patterns, visually representing the patterns, and navigating in the pattern space. The major contribution of three systems include:

- **Linear Pattern Detection**: This system allows users to visually examine the parameter space, i.e., the linear trend coefficient space, to discover linear trends and set appropriate thresholds, such as maximum tolerance and minimum coverage. The sampled parameter space shows where the 'good' linear patterns may exist and the user can interactively adjust the sample point, which is an extracted linear pattern. The preliminary results suggest that the system can facilitate discovering multiple coexisting linear trends and extracting more accurate trend using computational techniques after interactively removing the outliers that are outside the trend boundary. The user study shows that this system can better help the users discover the hidden linear model in the datasets, compared to the computational methods.

- **Visual Subgroup Mining**:

  The main contribution for this system is that we allow users to interactively submit subgroup mining queries for discovering interesting patterns and visually examine the mining result. Specifically, our system can accept mining queries dynamically, extract a set of hyper-box shaped regions called *Nuggets* for easy understandability and visualization, and allow users to navigate in multiple views for exploring the query results. I proposed a multi-layer structure to assist the user examine the patterns in different level of details. While navigating in the spaces, users can specify which level of abstraction they prefer to view. Meanwhile, the linkages between the entities in different levels and the corresponding data points in the data space are highlighted. The user study indicates that this system can better help the

users understand the mining result and identify interesting subgroups, compare to existing tabular knowledge representations.

- **Local Patterns Exploration and Anomaly Detection**: In this system, the patterns for the local sub-region with a focal point and its neighbors are computationally extracted and visually represented for examination. The extracted local pattern is used for sensitivity analysis. I designed a pointwise exploration method to allow users to examine the neighbors of a focal point. To discover anomalous patterns, the extracted local patterns are integrated and visually shown to the analysts. Users can discover the anomalies based on the distributions of global patterns. The user study showed that the designed local pattern view is better to assist the users understand the relationship between the selected focal point and its neighbors. It also helps the users more quickly identify interesting neighbors.

Table 6.1 gives a summary and comparisons of the three systems.

6.2 Contributions

The main features of the systems and contributions of my dissertation include:

- **Pattern extraction**: The main goal for my dissertation was to assist analysts computationally, visually, and interactively discover and extract interesting patterns, such as trends, clusters, and outliers from multivariate datasets. The proposed systems allow the users to mine different types of patterns and specify what kind of patterns they expect to extract, including the pattern type and parameters.

- **Pattern representation**: After the patterns are extracted according to the users' requirement, the next step is to visually represent each pattern to help the users understand each individual pattern, the relationship among patterns and how they are distributed in the pattern space. In the nugget browser system, I used star glyphs to visually represent each nugget and the layout strategy shows the relationships among different extracted patterns.

- **Pattern exploration**: Interactions are provided so that the users are able to explore in the pattern space. Since the pattern space is usually sampled or discretized, to discover more interesting data items in the pattern space, the exploration must be interactive. For example, in the linear pattern discovery system described in Chapter 3, we provide users a sampled model space, where users can select a single point and explore in the space.

- **Pattern refinement**: Users can refine their queries to extracted more appropriate patterns. Also, users can adjust each pattern to improve accuracy. For example, in the linear pattern discovery system mentioned in Chapter 3, users can adjust the discovered or computed linear trend in a model selection panel. The line width and

	Linear Pattern Detection	**Visual Subgroup Mining**	**Local Patterns Exploration**
Interestingness	modeling prediction	statistical detection prediction	sensitivity analysis anomaly detection
Shape of the subregion	areas between two hyper-planes	clustered cells hyper-box shaped nuggets	areas around the focal points
Pattern discovery (computational or user exploration)	explore in the model space tune the model	user involved subgroup definition statistical method	user involved neighbor definition sensitivity extraction statistical method
Pattern selection Connections between model and data	tune in a parallel coordinate view highlight the data points with color; distance mapping	select an item in one layer display the data involved when selecting	click an instance as a focal point highlight all neighbors with large glyphs
Data space display	Scatterplot matrices	Parallel Coordinates	Star Glyphs
Pattern space display	3 designed views	3 coordinated views	comparative display
Color mapping	distance to the trend	indicate the significance level	indicate the outlier factor
Position mapping	projection view	layout of nugget space	layout of local neighbors
Representation method	line graph histogram	star glyphs parallel coordinate	designed comparative method
Pattern adjustment	drag in a parallel coordinate view	tune cut-point positions tune target share range	change the coefficients
Evaluations	User-driven model selection	Visual representation of mining result	Visual local pattern representation

Table 6.1: A summary of the three proposed visual mining systems.

line color represent the goodness of the current trend. In the pointwise locl pattern exploration system, described in Chapter 5, users are allowed to customize the local pattern based on their requirements.

6.3 Future Work

In the future, I'd like to:

- Extend the parameter space exploration and visualization to other general models, such as non-linear forms. Extend this model extraction problem to other data mining tasks, for example, not only for regression, but also for discrimination and classification tasks.

- Add more interactions and more complex mechanisms for managing the user's discoveries in the Nugget Browser system, such as adjusting nugget boundaries with domain knowledge, as well as removing highly overlapping nuggets. Another extension is to use the extracted nuggets as evidence to verify the hypotheses about the casual relationship between the independent and target variables. Therefore, an evidence pool is a useful feature that can be supported in the future.

- Extend the pointwise local pattern exploration to support more types of patterns, such as distances. Interactively submitting queries for detecting interesting local pattern can also be supported in the future, for example, finding similar local patterns based on an interesting one.

- Continue to evaluate the systems with users. Longitudinal studies could be performed to analyse the learning curves of different systems.

- Since "nugget" denotes a subset of data or any interesting findings in multivariate datasets, this idea of knowledge discovery can be extended to a more general use. Some other visual analytic systems can be proposed, implemented and evaluated. These systems can assist users in computational and interactively extracting nuggets, visually representing each nugget and the relationship between the nugget and data for better understanding, as well as interactively adjusting the nuggets with user's domain knowledge. Some other potential future work includes: discovering other types of patterns, which are not mentioned in this dissertation, such as graph/structure based patterns or surprising patterns; discovering different patterns in subspaces or lower dimensional space projections; mixing nuggets of different types from different systems; supporting collaborative nugget-based analysis; as well as managing and comparing the findings from different parameter settings or data sources.

Bibliography

[1] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *SIGMOD '98*, pages 94–105. ACM, 1998.

[2] M. Ankerst. Visual data mining. In *in Ph.D. Dissertation, Mathematics and Computer Science, University of Munich*, 2000.

[3] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: An interactive approach to decision tree construction. In *Proc. 5th Int. Conf. on Knowledge Discovery and Data Mining (KDD 99)*, pages 392–396, 1999.

[4] M. Ankerst, M. Ester, and H.-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 179–188, New York, NY, USA, 2000. ACM.

[5] M. Ankerst, D. A. Keim, and H.-P. Kriegel. Circle segments: A technique for visually exploring large multidimensional data sets. In *Proc. IEEE Visualization'96*, 1996.

[6] D. Asimov. The grand tour: a tool for viewing multidimensional data. *SIAM J. Sci. Stat. Comput.*, 6(1):128–143, 1985.

[7] M. Atzmueller. Subgroup discovery. In *Künstliche Intelligenz*, volume 4, pages 52–53, 2005.

[8] Y. Aumann and Y. Lindell. A statistical theory for quantitative association rules. *J. Intell. Inf. Syst.*, 20(3):255–283, 2003.

[9] B. Becker, R. Kohavi, and D. Sommerfield. *Visualizing the simple Baysian classifier*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[10] C. Beilken and M. Spenke. Visual, interactive data mining with infozoom - the medical data set. In *Proceedings 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases, PKDD*, 1999.

[11] S. Berchtold, H. V. Jagadish, and K. A. Ross. Independence diagrams: A technique for visual data mining. In *Proc. Knowledge Discovery and Data Mining*, pages 139–143, 1998.

[12] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York, NY, USA, 1996.

[13] G. E. P. Box and N. R. Draper. *Empirical model-building and response surface*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[14] C. Brunk, J. Kelly, and R. Kohavi. Mineset: An integrated system for data mining. In *Proc. Conf. Knowledge Discovery and Data Mining*, pages 135–138, 1997.

[15] C. Collins and S. Carpendale. Vislink: revealing relationships amongst visualizations. *IEEE Trans Vis Comput Graph*, 13(6):1192–1199, 2007.

[16] Color brewer, obtained on february 6 2011. http://colorbrewer2.org.

[17] K. C. Cox, S. G. Eick, G. J. Wills, and R. J. Brachman. Visual data mining: Recognizing telephone calling fraud.

[18] U. Cvek, A. Gee, G. G. andP. Hoffman, K. Marx, D. Pinkney, M. Trutschl, and H. Zhang. Data mining of yeast functional genomics data using multidimensional analytic and visualization techniques. *Drug Discovery Technology*, 1999.

[19] M. C. F. de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *Journal of Computaional and Graphical Statistics*, 4(6):113–122, 1996.

[20] M. C. F. de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualizatoin and Computer Graphics*, 9(3):378–394, 2003.

[21] B. A. Dobson, A.J. *An Introduction to Generalized Linear Models*. Chapman and Hall, London, 2008.

[22] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[23] J. G. Dy and C. E. Brodley. Visualization and interactive feature selection for unsupervised data. In *In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 360–364, 2000.

[24] M. Ester, H. peter Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 96)*, pages 226–231. AAAI Press, 1996.

[25] U. Fayyad. Mining databases: Towards algorithms for knowledge discovery. volume 21, pages 39–48, 1998.

[26] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Advances in knowledge discovery and data mining. chapter From data mining to knowledge discovery: an overview, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[27] Y. Fua, M. Ward, and E. Rundensteiner. Navigating hierarchies with structure-based brushes. *Proc. IEEE Symposium on Information Visualization*, pages 58–64, 1999.

[28] A. L. G.A. Helt, S. Lewis and G. Rubin. Bioviews: Java-based tools for genomic data visualization. volume 8, pages 291–305, 1998.

[29] Google maps, obtained on january 21 2011. http://maps.google.com.

[30] Z. Guo, M. O. Ward, and E. A. Rundensteiner. Model space visualization for multivariate linear trend discovery. In *VAST '09*, pages 75–82. IEEE Computer Society, 2009.

[31] Z. Guo, M. O. Ward, and E. A. Rundensteiner. Nugget browser: Visual subgroup mining and statistical significance discovery in multivariate dataset. *IV2011: 15th International Conference Information Visualisation*, pages 267–275, 2011.

[32] J. Han and N. Cercone. Ruleviz: A model for visualizing knowledge discovery process. In *Proc. Int?l Conf. Knowledge Discovery and Data Mining (ACM SIGKDD '00)*, pages 244–253, 2000.

[33] J. Hartigan and B. Kleiner. Mosaics for contingency plots. In *Proc. 13th Symp. Interface*, pages 268–273, 1981.

[34] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas. Interactive data analysis: The control project. *Computer*, 32:51–59, August 1999.

[35] A. Hinneburg, D. A. Keim, and M. Wawryniuk. Hd-eye: Visual mining of high-dimensional data. *IEEE Computer Graphics and Applications*, 19:22–31, 1999.

[36] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. Dna visual and analytic data mining. In *VIS '97*, pages 437–441. IEEE Computer Society Press, 1997.

[37] H. Hofmann, A. Siebes, and A. Wilhelm. Visualizing association rules with interactive mosaic plots. In *Knowledge Discovery and Data Mining (ACM SIGKDD 00)*, pages 227–235. IEEE Computer Society, 2000.

[38] A. Inselberg. Multidimensional detective. In *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis '97)*, pages 100–, Washington, DC, USA, 1997. IEEE Computer Society.

[39] A. Inselberg and T. Avidan. Classification and visualization for high-dimensional data. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 370–374, New York, NY, USA, 2000. ACM.

[40] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *VIS '90: Proceedings of the 1st conference on Visualization '90*, pages 361–378, Los Alamitos, CA, USA, 1990. IEEE Computer Society Press.

[41] James allen jewelry online store dataset, obtained on may 19 2010. http://www.jamesallen.com.

[42] M. Jelovič, J. Jurić, Z. Konyha, and D. Gračanin. Interactive visual analysis and exploration of injection systems simulations. In *Proc. of the IEEE Visualization*, pages 391–398, 2005.

[43] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[44] D. A. Keim, M. Ankerst, and M. Sips. *Visual Data-Mining Techniques*, pages 813–825. Kolam Publishing, 2004.

[45] D. A. Keim and H.-P. Kriege. Visualization techniques for mining large databases: A comparison. *IEEE Transactions on Knowledge and Data Engineering*, 8:923–938, 1996.

[46] D. A. Keim, C. Panse, and M. Sips. Information Visualization: Scope, Techniques and Opportunities for Geovisualization. In J. Dykes, A. M. Maceachren, and M. J. Kraak, editors, *Exploring Geovisualization*, pages 23–52. Elsevier, Amsterdam, 2005.

[47] R. Kohavi and D. Sommerfield. Targeting business users with decision table classifiers. In *Proc. Conf. Knowledge Discovery and Data Mining (ACM SIGKDD'98)*, pages 249–253, 1998.

[48] A. Madansky. The fitting of straight lines when both variables are subject to error. *Journal of the American Statistical Association*, 54(285):173–205, 1959.

[49] Minnesota Department of Transportation. Mn/DOT traveler information. http://www.dot.state.mn.us/tmc/trafficinfo/index.html/, accessed on Feb. 16, 2009.

[50] E. Moura and D. G. Henderson. *Experiencing geometry: on plane and sphere*. Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[51] O. Niggemann. Visual data mining of graph-based data. dissertation. In *Department of Mathematics and Computer Science. University of Paderborn, Germany*, 2001.

[52] F. Oellien, W.-D. Ihlenfeldt, and J. Gasteiger. Infvis - platform-independent visual data mining of multidimensional chemical data sets. *Journal of Chemical Information and Modeling*, 45(5):1456–1467, 2005.

[53] W. Ribarsky, J. Katz, F. Jiang, and A. Holland. Discovery visualization using fast clustering. *IEEE Computer Graphics and Applications*, 19:32–39, 1999.

[54] J. A. Robinson and T. P. Flores. Novel techniques for visualizing biological information. In *Proceedings of the 5th International Conference on Intelligent Systems for Molecular Biology*, pages 241–249. AAAI Press, 1997.

[55] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

[56] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, Gatelli, D. Gatelli, M. Saisana, and S. Tarantola. *Global Sensitivity Analysis: The Primer*. John Wiley and Sons, 2008.

[57] J. W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.*, 18:401–409, May 1969.

[58] S. J. Simoff, M. H. Böhlen, and A. Mazeika. Visual data mining. chapter Visual Data Mining: An Introduction and Overview, pages 1–12. Springer-Verlag, Berlin, Heidelberg, 2008.

[59] J. Symanzik, G. A. Ascoli, S. S. Washington, and J. L. Krichmar. Visual data mining of brain cells. *Computing Science and Statistics*, pages 445–449, 1999.

[60] UC Irvine Machine Learning Repository. Mammographic Mass Data Set, obtained on december 8 2010. http://archive.ics.uci.edu/ml/datasets/Mammographic+Mass.

[61] J. Vesanto. Som-based data visualization methods. In *Intelligent Data Analysis*, volume 3, pages 111–126, 1999.

[62] J. Vesanto. Using soms in data mining. In *Licenciates thesis, Helsink Univ. of Technology*, 2000.

[63] W. Wang, J. Yang, and R. R. Muntz. Sting: A statistical information grid approach to spatial data mining. In *VLDB '97*, pages 186–195. Morgan Kaufmann Publishers Inc., 1997.

[64] M. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. *Proc. IEEE Visualization*, pages 326–333, 1994.

[65] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten. Interactive machine learning - letting users build classifiers. In *Working Paper, Dept. of Computer Science, Univ. of Waikato*.

[66] P. Wong. Visual data mining. *IEEE Computer Graphics and Applications*, 19(5):20–21, 1999.

[67] D. Yang, E. A. Rundensteiner, and M. O. Ward. Analysis guided visual exploration of multivariate data. In *Proc. VAST 2007. IEEE Symposium on*, pages 83–90, 2007.

[68] J. Yang, M. Ward, and E. Rundensteiner. Hierarchical exploration of large multivariate data sets. In F. Post, G. Nielson, and G.-P. Bonneau, editors, *Data Visualization: The State of the Art 2003*, pages 201–212. Kluwer, 2003.