# A Model-driven Visual Analytic Framework for Local Pattern Analysis

Kaiyu Zhao

A Dissertation

Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Doctor of Philosophy

in

Computer Science

by

---

February 2016

APPROVED:
Professor Matthew O. Ward, Advisor (Posthumously)

---

Professor Elke A. Rundensteiner, Co-Advisor

---

Professor Joseph E. Beck, WPI

---

Professor Xiangnan Kong, WPI

---

Professor Jimmy Johansson, University of Linköping

## Abstract

The ultimate goal of any visual analytic task is to make sense of the data and gain insights. Unfortunately, the process of discovering useful information is becoming more challenging nowadays due to the growing data scale. Particularly, the human cognitive capabilities remain constant whereas the scale and complexity of data are not. Meanwhile, visual analytics largely relies on human analytic in the loop which imposes challenge to traditional human-driven workflow. It is almost impossible to show every aspect of details to the user while diving into local region of the data to explain phenomenons hidden in the data. For example, while exploring the data subsets, it is always important to determine which partitions of data contain more important information. Also, determining the subset of features is vital before further doing other analysis. Furthermore, modeling on these subsets of data locally can yield great finding but also introduces bias. In this work, a model driven visual analytic framework is proposed to help identify interesting local patterns from the above three aspects. This dissertation work aims to tackle these subproblems in the following three topics: *model-driven data exploration*, *model-driven feature analysis* and *local model diagnosis*. First, the *model-driven data exploration* focus on the problem of modeling subset of data to identify the co-movement of time-series data within certain subset time partitions, which is an important application in a number of domains such as medical science, finance, business and engineering. Second, the *model-driven feature analysis* is to discover the important subset of interesting features while analyzing local feature similarities. Within the financial risk dataset collected by domain expert, we discover that the feature correlation among different data partitions (i.e., small and large companies) are very different. Third, *local model diagnosis* provides a tool to identify interesting local regression models at local regions of the data space which makes it possible

for the analysts to model the whole data space with a set of local models while knowing the strength and weakness of them. The three tools provide an integrated solution for identifying interesting patterns within local subsets of data.

## Acknowledgements

I would never have been able to finish my dissertation without the guidance of my committee members, and support from my family, especially my wife who contributed a tremendous amount of her time to support me.

I would like to express my sincere gratitude to my advisor, Dr. Matthew O. Ward, who guided and mentored me with great patience. His persistent dedication to work and research motivates me to make progress on my dissertation research. His spirit vigorously influences me to face any difficulties positively at times before and after he passed away.

I would like to thank my co-advisor, Dr. Elke A. Rundensteiner, who energetically replenished my knowledge and refined my work. Her diligence towards any seemingly trivial issues always leads to non-trivial research questions worth further investigation.

My thanks also go to Dr. Joseph E. Beck, who serves as my first year academic advisor, reader of my qualifier examination, and my committee member. He opened a door for me which lead to a whole new world of advanced studies. Working with him prepared me with a great learning methodology.

My thanks go to Dr. Jimmy Johansson. He pointed me to interesting visualization research topics that benefits my researching problems.

My thanks go to Dr. Xiangnan Kong, whose suggestions and comments inspired me in machine learning related research.

I want to thank everyone in my committee for their time, encouragement and valuable ideas while I was working on this dissertation.

My thanks go to all members of Xmdv, ISRG and DSRG who made suggestions to my work.

Finally, I appreciate the financial support from NSF that funded the research discussed in this dissertation.

# Contents


**1 Introduction** **1**

1.1 Background . . . . . 1

1.2 State of the Art . . . . . 2

1.3 Proposed Approach for Local Pattern Analysis . . . . . 6

1.3.1 Model-driven Data Exploration . . . . . 7

1.3.2 Model-driven Feature Analysis . . . . . 9

1.3.3 Local Model Diagnosis . . . . . 11

1.4 Contributions of this Dissertation . . . . . 12

1.4.1 Model-driven Data Exploration: . . . . . 12

1.4.2 Model-driven Feature Analysis: . . . . . 13

1.4.3 Visual Guided Model Diagnosis: . . . . . 13

1.5 Dissertation Outline . . . . . 14

**2 Model-driven Data Exploration** **15**

2.1 Preliminaries of Data Patterns and Models . . . . . 16

2.1.1 Drift Model . . . . . 19

2.1.2 Seasonal Model . . . . . 19

2.1.3 Uncertainty Model . . . . . 20

2.2 Proposed MaVis Framework . . . . . 21

2.2.1 Data Space . . . 22
2.2.2 Model Space . . . 25
2.2.3 Model Relation Space . . . 29
2.2.4 Nugget Space . . . 30
2.3 Evaluation . . . 35
2.3.1 Case Study: Stock Price Co-movement . . . 35
2.3.2 User Study Design . . . 38
2.3.3 User Study Result . . . 41
2.4 Related Work . . . 43
2.5 Summary . . . 45

**3 Model-driven Feature Analysis** **46**
3.1 FeaVis Workflow . . . 47
3.2 Feature Clustering . . . 50
3.2.1 Correlation Coefficient . . . 51
3.2.2 K-th Central Moment . . . 52
3.2.3 Cross Entropy . . . 53
3.2.4 Automatic Weighted Feature Clustering . . . 54
3.2.5 Cross Metric Similarity View . . . 57
3.3 Feature Ranking . . . 59
3.3.1 Diversified Feature Ranking . . . 59
3.3.2 Feature Cluster Drill-down View . . . 63
3.4 Feature Pruning . . . 66
3.4.1 Partition Based Redundancy Pruning . . . 66
3.4.2 Redundancy Inspection View . . . 68
3.5 Evaluation . . . 71

3.5.1 Data Description . . . 72
3.5.2 Case Study: Representative Financial Variables . . . 72
3.5.3 Comparison to Empirical Studies . . . 74
3.6 Related Work . . . 76
3.7 Summary . . . 79

**4 Local Model Diagnosis** **81**
4.1 Model Complementarity Visualization . . . 85
4.1.1 Goodness Measure . . . 85
4.1.2 Point-wise Comparison . . . 87
4.1.3 Stacked Binned Summary View . . . 87
4.2 Model Diversity Visualization . . . 90
4.2.1 Isolating Multiple Local Models . . . 90
4.2.2 Mutable Partitions . . . 92
4.2.3 Partition Layout and Representation . . . 92
4.3 Model Representativity Visualization . . . 94
4.3.1 Representative Trend . . . 95
4.3.2 Interactive Local Trend Aggregation . . . 96
4.3.3 Aggregation Quality Loss . . . 97
4.4 Evaluation . . . 99
4.4.1 Case Study: Linear Models of Bankruptcy Risks . . . 99
4.4.2 User Study for Evaluating Model Fit . . . 103
4.5 Related Work . . . 104
4.6 Summary . . . 106

**5 Conclusion and Future Directions** **107**
5.1 Conclusion: . . . 107

5.2 Future Directions: . . . . . 108

# List of Figures

1.1 Conceptual Picture of this dissertation work. . . . . 2

1.2 Basic data reduction techniques for visualizing large scale data. . . . . 3

1.3 Pairwise variable relationship visualization. A conceptual picture of several typical patterns of 2-D variable relationship. The displaying space of this figure is a MDS layout of 2-D scatter plots [WAG05] where the positioning of one particular plot is determined by the pattern it shows. . . 5

1.4 Confusion matrices of EnsembleMatrix [TLKT09]. Multiple classifiers are shown in thumbnails on the right. The matrix on the left shows the confusion matrix by aggregating classifiers using weights. . . . . 6

2.1 Comparison of two binning strategies for collection of time series. The binning method may count every data point (a) or count the number of time series (b). Counting every data point highlights grid cells that have multiple occurrences of data points but with only one time series (c). . . . 19

2.2 *Time line movement* view (b) presents a collection of 250 time series where x-axis represents the time progression and y-axis is the normalized price values ranging from 0 to 1. The darker region in the view at around October 2008 shows that the majority of the companies were at relatively low price values. The line chart view (a) presents the data with the same normalization method (view rendered within Excel). . . . . 22

2.3 Two constraint boxes are placed to reveal companies that fell (a) and rose (b) during the 2008 crisis. Compared to the view in Fig 2.2b, we see that most ( 70/ 100) of the prices move with such behavior. The color schema range is adjusted based on the maximal count of all the grid cells by default. 23

2.4 Drift abstraction of a collection of 32 time series objects. a) The default color encoding which represent the count of time series in each bin. b) Filter operator selects time series lower than the risk neutral zone. The color encoding represents the count of selected time series. c) Link the selected time series in space b back to original data space. The leftmost histogram shows the overall drift of the time series over the selected time span (2006 and 2007). The histograms to the right with white background show the local drift of each company at the granularity of 6 months in each view. In these sets of views, we observe several interesting patterns. (1) Most companies stay in the risk neutral zone which is the longest bar in all the histograms while many companies fell down at the end of 2007. (2) We can also observe an outlier time series (Apple) that grows exceptionally. (3) Linking from the model space view (highlighted rectangles in leftmost rectangle of b) to the time line movement view reveals an overall falling pattern with high density towards the end of 2007 in (c). . . . . 25

2.5 Time series similarity in the drift model space. The leftmost *bar code view* visualizes the overall drift tendency of the selected time series where each line corresponds to one time line. The 5 bars to its right visualize the local drift. . . . . 26

2.6 Model similarity analysis view. a) A brushed co-moving drift pattern begins since about July 2006. b) The darker color bins show a high correlation between different time intervals. The drift estimate of bins in (a) and that in (b) are at relatively the same value range. It shows the drift of co-moving patterns is quite consistent over time. c) A high degree of volatility is shown. d) A long seasonal cycle is represented. . . . . 30

2.7 The view represents a collection of time series with a co-moving trend that is identified in the first time interval indicated by the green box plot (a). However, the co-movement pattern of the same group became gradually diverging over time and reached peak during the last time interval (greatest variance indicated by the height of bars) (e). From a long term perspective, the co-movement pattern that is identified in the green model space is more consistent across the three model types at time interval (f) compared to the other local intervals (a-e). . . . . 31

2.8 The views show a interactive exploration process for co-movement pattern investigation. a) The overall drift pattern is presented as heatmap view. b) Filtered results are shown after a range query is submitted. In the view to the right, co-moving patterns are linked via color encoding. c) When the collection of growing time series are selected the corresponding risk of this collection is linked to other portion of the views such as (d) (e) and (f). d) The boxes have darker colors which indicates higher correlation. e) The lighter color there shows lower correlation. f) The pattern is also showing some degree of correlation but at high dispersion which means the collection is less likely co-moving. . . . . 34

2.9 The first row (from left to right) shows the summary statistics of the selections in Fig 2.8d), e) and c). The second row shows the same glyphs with focus on a reference glyph for comparison. The similarity score is calculated between the reference glyph and the other glyphs (second row) and then the similarity score is rendered as alpha value of the glyph color. 36

2.10 The chosen design of the views in question 1A and question 2A requires less time for discovering the pattern of interest. The two glyph views tested in question 3A require relatively the same amount of time. However, the chosen design has better accuracy as discussed in Sec 2.3.3. . . . 40

2.11 Each question B has 5 options (x axis) a subject may choose from. Option 1 to 3 (Sec 2.3.2) for question B are supported by our system and the subject may dig further to discover more insights. Option 4 is *Don't know* which means the subject has no more questions. Option 5 is *Other* and the subject may have additional questions to query the system but we do not yet support those. Bars with 3 different colors represent three views we are evaluating (1B:time line movement view, 2B:model similarity view, 3B:nugget analytic view). Y axis represent number of subjects who chose the corresponding option. Based on the result, few subjects chose option 5 indicating the framework covers most their further needs initiated from the given 3 questions. . . . . 42

2.12 Accuracy comparison between our choices and alternative options. Y axis shows the percentage of subjects who correctly recognized the pattern in the design space. X axis lists the design choices we have for the three views. 43

3.1 The overall workflow of the FeaVis system. The top 3 components are model-driven algorithmic methods that search the most descriptive subset of features based on given metrics automatically. The bottom 3 components are interactive visual support that help refine and interpret the automatic processes. . . . . 48

3.2 The view shows a comparison between the 3 default ranking metrics (left) and 3 user metrics generated by combining the 3 default metrics (right). The user metrics are generated by using different weight combinations, in this case [0.5,0.3,0.2], [0.1,0.8,0.1] and [0.4,0.2,0.4] respectively. The feature on top is the focused feature and it has similarity score of 1 to itself. Other features are ranked based on similarities to the focus feature using different metrics. The length of bars represent the similarity score. (AT: total assets; LSE: leverage; LogAT: log total asset; LT: total liability; SALE: total sale; GP: gross profit; MKVALT: market value; XSGA: general expenses; DLTT: long term debt; XINT: interest expenses.) . . . . 58

3.3 The detailed view of a cluster of features. The column represents a feature, and for each column the color of a grid indicates how far this feature is away from its neighbors. The first column is automatically selected as a representative of this group (long rectangle). The small red selection box to the right in the view is a cursor over selection which shows more information about that particular neighbor. . . . . 62

3.4 Cluster view of 45 features in 10 groups, including one single element group represented by a cyan rectangle. The group can be selected/unselected and the selections are marked with small red boxes. The black circle over the group indicates a marked focus group by an analyst, the details of the focus group are displayed in a different view, shown in Figure 3.3. . . . . 64

3.5 (a) A relatively large feature group with high in-group similarity, indicated by the relatively low average distance, as well as low variances. It indicates the large group of features are very similar to each other. Thus the redundancies in this group is significant. (b) Based on the same reason, b shows high intra cluster similarity but it is a much smaller cluster. (c) It is a relatively large group with low in-group similarity. The confidence of removing redundancies in this group using automatic methods is less for the group on the right. . . . . 65

3.6 Illustration of process for partitioning on two features. The bin size is 3 and the number of bins is 2. . . . . 68

3.7 The analysts may examine the stability of the feature similarity across every partition. In this view, each histogram view represents the stability of one feature vs the others within a cluster. The horizontal red line indicates the global similarity between the given feature and the others. The label underneath each histogram represents the name of the given feature. The x-axis of each histogram represents partitions generated on the given feature arranged from low value to high value from left to right. The y-axis represents the degree of redundancy from low to high. The shape of the histogram represents the stability based on how close the bars are to the red base line. . . . . 70

3.8 This view shows the features selected by global redundancy measure when the analyst finishes adjusting the number of groups. This selection is done without conducting any local redundancy analysis. . . . . 74

3.9 This view shows the features selected by local redundancy measures over a subset of data points. This view is generated after the analyst brushes the partitions of interest. . . . . 75

4.1 The two plots show that the two models displayed by the line trend oppose each other in terms of bias. $Model_1$ has the tendency to underestimate and $Model_2$ tends to overestimate when the total asset grows. The y-axis shows the goodness of fit (residuals). The x-axis is the value of total assets (one of the independent variables). DLTT: Total long-term debt; LEV: Leverage; MKVALT: Market value . . . . . 84

4.2 The plots represent how the linear relationship between two variables can differ when considering different partitions of data points. From a domain expert point of view, both high return and low return companies have relatively high risk; intermediate return (fluctuate around 0) companies tend to follow a trend whose risk is reversely proportional to the return. . 85

4.3 Integrated analysis framework with 3 stages. **1)** Variables are ranked by their relevance to the dependent variable. The scatterplot (a) shows the relationship between a selected independent variable and the dependent variable. The global models built by the analysts are listed in (b). Model complementarity is presented in (c) for refining a model in (b). **2)** Local models can be derived from a selected global model and are presented in (d,e). **3)** The local models are grouped and summarized in a hierarchy (f). 86

4.4 A candidate model LEV complements the to-be-refined model DEBTTA (in the yellow box). The y-axis represents the error spread of two models. Positive (Negative) values suggest bias towards underestimate (overestimate). The x-axis represents local partitions where the errors are estimated. The theme river design [HHN00] represents the residuals of the to-be-refined model. The red vertical lines represent the residuals of a candidate model (usually a univariate model). . . . . 89

4.5 The x-y position of any cell in the grid view (a) is determined by the lower (x) and upper (y) percentile threshold of a data partition. The relationship between the x-y position and the partition boundary is shown in (b) and is indexed as in (c,d). Each cell is colored by the fitness of a local model in it. The diagonal and the orthogonal direction in (c) indicates two ways a data partition may change to another: expanding (add more data points) and shifting (add data points at one end and remove at the other). An time chart display (Fig 4.4b) of (a) is transformed from (a) by the sequence in (d) where the main diagonal is walked from top left first followed by the second diagonal above it. The walk continues till the right top corner. . . 91

4.6 Visualize the degree of diversities. It shows that the local models isolated by partitioning on DLTT (a,b) have more diversity over the local models isolated by partitioning on ARChange (c,d). ARChange: Account Receivable Change . . . . . 94

4.7 Visualize the coverage (cells with red outline on the left) of a selected cluster of data partitions (selected node marked with red rectangle on the right). . . . . 97

4.8 Visualize the coefficient vector (red horizontal bars in the icicle plot) of the linear trend in the highlighted data partition (left). The red text shows the value of the coefficients and the name of variables. The color scale shows the relative goodness of local models in a corresponding partition. . 98

4.9 A case study for modeling risk. a) A ranking list of independent variables. b) Scatterplot of a selected independent variable and the dependent variable. c) A list of built models. d, e) Complementarity analysis. . . . . 99

4.10 A case study for modeling risk. f), g), h) and i) Local model diversity analysis. . . . . 101

4.11 A case study for modeling risk. j), k), l) and m) Model representivity analysis. . . . . 102

# List of Tables

3.1 Example of similar features for feature *total assets*. By default the aggregation weight is 0.333 for each metric and the similarity is normalized to (0,1). . . . . 57

3.2 Explanations of features . . . . . 72

3.3 The mark "x" indicates selection of that feature. The numbers in the last column are the measures of correlation $(1-|\rho|)$ between the selected feature and unselected features in a group. NA means there is no such feature available in our dataset. . . . . 76

4.1 Model specific metrics for quality evaluation . . . . . 83

4.2 User study accuracy results based on 3 questions. . . . . 104

# Chapter 1

# Introduction

## 1.1 Background

Visual analytics nowadays has to deal with increasingly large scale data. Analysts have to deal with larger scale of data than ever, in terms of higher volume and dimensionality. The significant bottleneck for large-scale visual analytics is the **human element** within the analytic workflow [WSJ+12]. As data scale continue to grow rapidly, the human cognitive abilities remain constant. To tackle the large scale data analytics problem, numerous data models are created to discover and extract useful information. However, to diagnose and fine tune the models generated by various of machine learning techniques tends to be a challenging problem. It involves a long tedious process of data engineering which is based on trial and error. To facilitate the such tasks, this dissertation focuses interactive model-driven visual analytics on three tasks, namely, *model-driven data exploration*, *model-driven feature analysis*, and *local model diagnosis*. Data analytic activities often involve the three tasks and they complement each other and serve the same purpose: interpret the data and make use of the knowledge gained from the data (Fig 1.1). The *model-driven data exploration* utilized machine learning models to capture interesting aspects

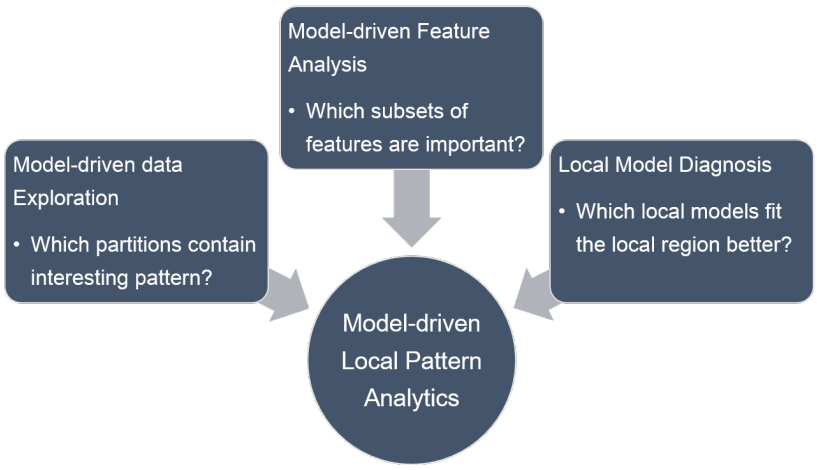


Figure 1.1: Conceptual Picture of this dissertation work.

of the data and enables analysts to compare and contrast patterns identified by different models. The *model-driven feature analysis* captures feature similarity and allow analysts to compare the correlations between different features discovered among data partitions. The *local model diagnosis* help analysts to identify strength and weakness of local models which are generated based on local subsets of data. The main focus of all these work is to visually support and guide analysts while they perform the above three tasks.

## 1.2 State of the Art

**Model-driven Data Exploration:** To alleviate the cognition load, data are often processed in a data reduction pipeline involving *binning*, *filtering*, *sampling*, *summarizing* and other steps [LJH13] (Fig 1.2). Such a data reduction process is usually a non-trivial task. The process has to capture the ”interestingness” of the data to provide an overview of the data space based on some standard. However, the standard can often only be determined by analysts after they ”see” the ”interestingness” of the data. Analysts often goes into a loop of generating hypothesis and verifying it via trial and error [Tuk77]. Unfortunately, this process sometimes does not only take significant amount of time given the

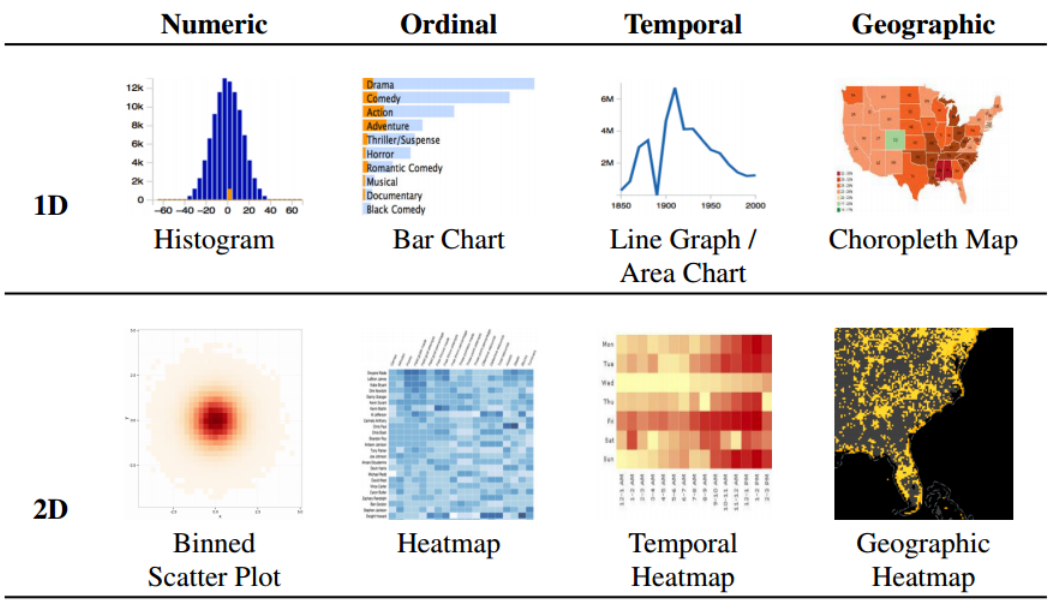


Figure 1.2: Basic data reduction techniques for visualizing large scale data.

complexity and the growing scale of data nowadays, but it also can be ineffective without appropriate visual support. Meanwhile, given the available machine learning models, analysts have the option of taking advantage of existing techniques to model the main characteristics of the data space and gain insight [GNRM08]. However, these work usually do not support multi-model comparison and analysts may still not know if they are using the right technique to approach their problem. This dissertation work provides visual guided modeling to capture the main characteristics of time series data for co-movement pattern discovery. Multiple time series models are integrated for analysts to compare and experiment.

**Model-driven Feature Analysis:** Visualizing a multi-variate dataset can be challenging due to "curse of dimensionality". The pairwise and/or higher order relationships between a number of features can be overwhelming. Visualizing such dataset without proper optimization usually leads to cluttered and ineffective display which can hardly lead to any useful insight [PWR$^+$]. Most visual analytic techniques are made well working for 4

or 5 dimensions but they are less effective for hundreds or more dimensions. Optimization strategies such as dimension reordering [JJ09] or pairwise summarization [WAG05] (Fig 1.3), often times reveal the important visual structures to analysts by filtering out the less interesting ones. Some of these optimization techniques are specific for a type of visualization (i.e., scatterplots) because the optimization process takes account of view-specific properties such as whether a point cloud in a scatter plot view is apparent to human eyes. These optimization techniques are helpful once the analyst has decided which visualization techniques to use. However, deciding on the appropriate view type for a given task type is a non-trivial problem in itself. Especially when analysts have no clue about the characteristics of the features such as the data types and the data distributions. This work instead integrates three feature similarity metrics (i.e., correlation, cross entropy and distribution similarity) and clustering models to help discover redundant features in the data space. Additionally, analysts is able to choose local partitions and identify feature similarities for a data partition of interest.

**Local Model Diagnosis:** Dozens of **evaluation metrics** have been proposed for visual quality measure [CWRY06, BTK11]. The visual quality metrics primarily focus on the quality of the display rather than that of data in general. For example, metrics for identifying views that are great for scatter plot projections do not necessarily help identify high order linear trends. Moreover, few metrics are designed to measure and diagnosis the quality of data abstraction and summarizations that are generated computationally by machine learning models in terms of "fitness". Furthermore, metrics are needed for analysts to understand the landscape of the data space. For example, *precision and recall* curves are indicators for diagnosing classification models. Interactive tools [TLKT09] (Fig 1.4) are also developed to support ensemble classifier diagnosis. However, most of these metrics are able to measure the quality of visual representations or data abstractions are global in nature. They are less effective for identifying local patterns (e.g., Simpson's Paradox)

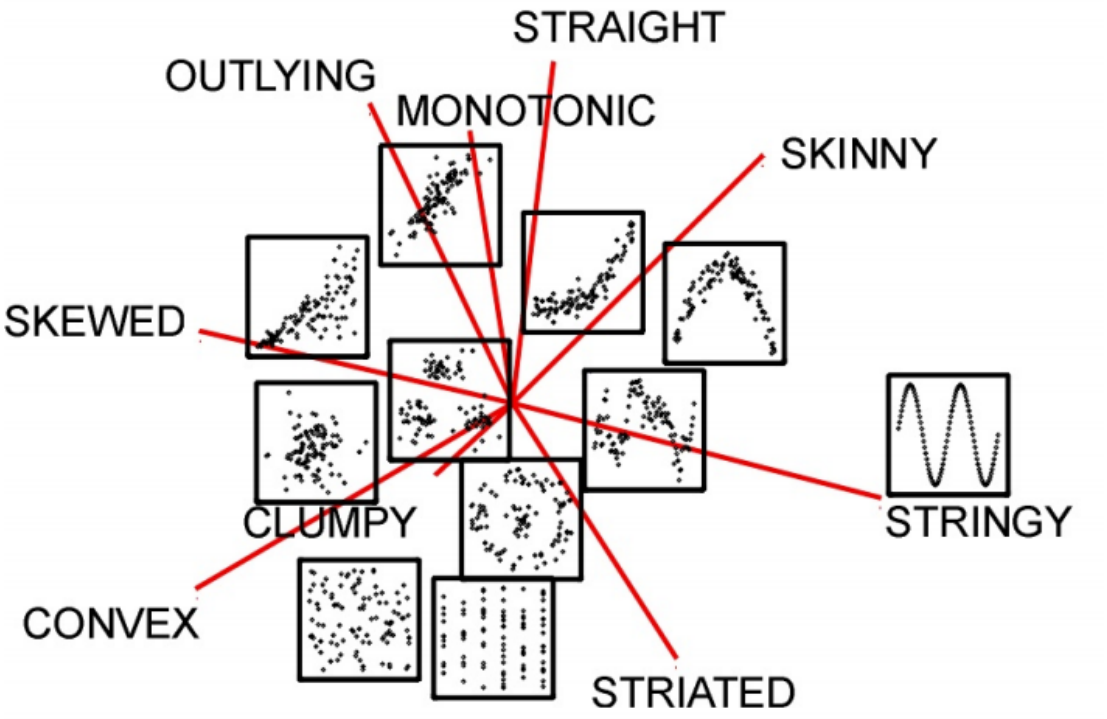


Figure 1.3: Pairwise variable relationship visualization. A conceptual picture of several typical patterns of 2-D variable relationship. The displaying space of this figure is a MDS layout of 2-D scatter plots [WAG05] where the positioning of one particular plot is determined by the pattern it shows.

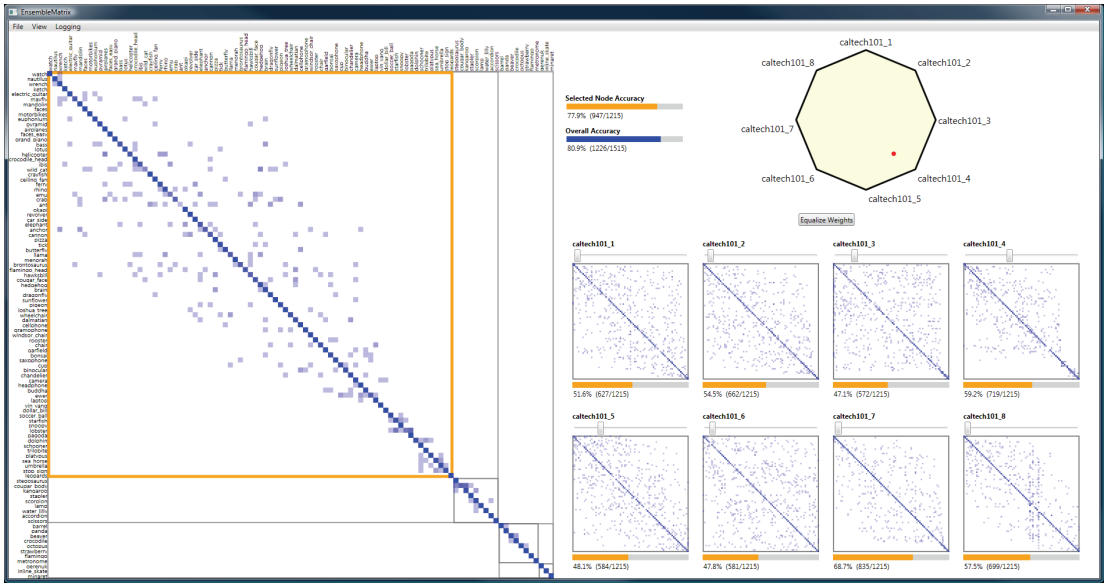


Figure 1.4: Confusion matrices of EnsembleMatrix [TLKT09]. Multiple classifiers are shown in thumbnails on the right. The matrix on the left shows the confusion matrix by aggregating classifiers using weights.

described by local models that are generated by subset of the dataset. In this work, three metrics are proposed to visualize and measure the goodness of regression models. They are designed to reveal local models of interest that fit the data well. Additionally, analysts may identify complement local models that can improve performance of others.

## 1.3 Proposed Approach for Local Pattern Analysis

To address the above challenges, a model-driven visual analytic framework is proposed and applied to the three areas of interest: *model-driven data exploration*, *model-driven feature analysis* and *local model diagnosis*. Henceforth, the ”model” in this work refers to machine learning models that are high level abstractions of the input data. For example, a linear trend model is an abstract representation of the underlying data that follows a certain linear trend. There are three main components for the model-driven approach. First, the models are used to summarize and describe large scale datasets to address the data

exploration problem. Second, the models can also be used to describe the relationships between different features to facilitate the feature selection process. Third, the models are then diagnosed, interpreted and refined in a visual environment to further interpret the landscape of the data space particularly the local subsets of interest. The model diagnosis aims to provide more insight by helping analysts to analyze local pattens captured by local models. All the three topics are collaborated work with a domain expert who is a Professor at school of business at Worcester Polytechnic Institute. The design of systems in this work is mainly motivated by the cognition limitations humans have [WGK10], such as limited ability to differentiate multiple colors on the screen. Therefore we prefer visualizing aggregated results (e.g., heatmap and histogram) to showing raw data to the user.

### 1.3.1 Model-driven Data Exploration

For the first task, a visual analytic tool called MaVis is proposed that integrates multiple machine learning models with a plug-and-play style to describe the input data. The data can often be processed in a data reduction pipeline involving binning, filtering, sampling, summarizing and other variations [LJH13]. Then analysts start to perform user-driven *exploratory data analysis* tasks. In this work, we provide model driven analytics such as model summarizations (clusters and trends) as well as data binning strategy. While investigating the co-movement patterns of time series dataset, this part of work aims to answer the following questions:

- What time intervals contain interesting co-movement patterns?
- What time-series model can I use to capture the co-movement?
- Which model is more interesting?

- What are the relationship between the models I use?

To answer these questions, we propose a plug-and-play visualization framework that integrate multiple machine learning models to summarize the interestingness of the raw data with four analytic spaces, data, models, model relationships or user queries. The models in MaVis are compact descriptions of the raw data such as clusters, trends and others. They are visualized and presented in a derived *model space* to provide *compacted representation* (e.g., cluster radius, slope and etc.) of the original raw data. The cognitive load can be significantly reduced by using machine learning models that lead to very compact descriptions. For example, 1 million data points can be effectively reduced to $k$ clusters ($k \ll 1$ million) in the *cluster model space* so that the analyst can have a grasp of the underlying data space.

This work includes a design of visual distinctions for the model descriptions, so that analysts can compare the models swiftly and determine which model to use for further exploration. MaVis incorporates 3 commonly used models and a higher level analytic space, namely, *model relation space*, to support such comparison activities via linked views. For example, to determine whether linear or non-linear trends are more appropriate to describe the underlying data, an analysts may want to compare the two models and decide which model type reveals more interesting patterns.

As discussed in [ZWRH14, MP13], the description of a model (e.g., slope of trend) is also determined by the data partition of the data space. For example, the trend slope of this year's data may be different from that of last year's. MaVis provides analysts the capability of managing and comparing their discoveries in a *nugget space* to keep track of the findings of an analyst. A nugget contains a subset of the points of interest and then summarize it for future analysis. For example, when an analyst identifies two clusters in two different data partitions, the *nugget space* maintains summaries of such observations which may lead to other discoveries such as overlap of two clusters.

### 1.3.2 Model-driven Feature Analysis

For the second task, the dissertation work primarily discusses the proposed visual feature exploration tool called FeaVis. It uses clustering techniques and feature similarity metrics to explore the feature spaces for selecting features of interest. Feature analysis is useful for reducing the scale of analysis by focusing the analysis on a subset of features [IMI+10a]. Feature analysis can be both expensive and difficult. To overcome such difficulties, dozens of techniques have been proposed to automate the feature selection process by considering feature similarities [MMP02, YL04, PLD05]. Since most automated feature selection processes are black-box approaches by nature, it is challenging to intervene and understand them. Furthermore, the designs are often based on specific algorithms not applicable in general for the exploratory analysis of features. An analyst may have questions to ask during analyzing the features of her data:

- Is this feature selection metric applicable to my problem?
- Why are my preferred features not picked by this selection algorithm?
- Are there any alternative methods I can use instead?
- Which features are correlated at which partitions?

To answer these questions, a multi-metric system is adopted in this work. Each metric measures the similarity of features from a particular aspect. Then the features are clustered into feature groups. This work integrates multiple metrics including pearson correlation, distribution similarity [kld] and cross entropy [DBKMR05].

To support feature similarity discovery and selecting most representative features, the metrics are combined to generate an aggregated measure which can be refined over the metric analytics. Specifically, for any specified feature, there can be different sets of features and each set corresponds to a particular metric. An analyst may then fine tune the

aggregation process to recompute the clustering result based on the multiple sets depending on which set of similarity relationship is more interesting determined by an analyst. For example, for a resort analytic dataset the feature set ⟨ *temperature, number of visitors* ⟩ is more interesting in the summer while the feature set ⟨ *snow fall, number of visitors* ⟩ is more interesting during the winter. Then the less interesting similarity metric can be deemphasized with a lower weight and vice versa.

Next, a ranking schema is designed to select the top-k most descriptive features within each group using a diversifying strategy. The features are first clustered into feature groups with a predefined aggregated similarity metric. Then k features are selected from each group for further inspection. To diversify the selection, the features are sorted based on a priority order of adjacent features being similar to each other. The selection then picks k dissimilar features from the feature group.

Furthermore, FeaVis also provides a drill-down functionality that an analyst is able to examine feature selections for different data subsets. A finer resolution analysis may involve investigating feature spaces in a subset of the original data space. According to the Simpson-Paradox [Wag82], local relationships between two variables can be totally distinct from the global relationship. For example, two redundant features (e.g., traffic jam and accidents) may be non-redundant for a given subset (e.g., traffic data around Los Angeles). The traffic jam and number of accidents may usually explain each other, but it is hardly true in a local region such as Los Angeles as there are always traffic jam there. To investigate this phenomena, the FeaVis system provides a partition importance view to direct analysts to the partitions of interest where a very different selection of features may be compared to the globally selected features.

### 1.3.3 Local Model Diagnosis

For this third task, a visual supported model diagnosis tool called LoVis (local pattern visualization) is proposed [ZWRH14]. The primary focus of this tool is to visually investigate how well a data abstraction method describes the dataset. For example, using a linear model to describe a dataset can potentially be inaccurate if there is any non-linearity or multiple linear segments in the dataset [MP13, ZWRH14]. Understanding the quality of the data modeling process facilitates the diagnosis of models in terms of fitness in describing the landscape of the underlying data space. The model-driven quality evaluation is analogous to visual quality measure techniques [CWRY06] where information loss caused by data transformation and mapping process is estimated. Generally, the info may also be lost or distorted by data reduction or data abstraction processes. For instance, the linear models can be used to describe the linear trends in a dataset with error and bias. The quality metrics allow analysts to summarize the main characteristics of the data with confidence.

While using model to summarize a data set, analysts may ask these questions before being able to confidently use the model to communicate ideas or report findings extracted from the data.

- Is my model accurately describing the whole data space?
- Does the model bias over a certain subset of the data?
- Is there any part of data that has very high error?

To help answer these questions, this work proposes a set of local measures and a mechanism to form models locally about certain interesting data subsets. We define three metrics for visualizing and measuring the quality of linear models particularly taking account for local patterns of the trends.

First, *model complementarity* is defined to find models that complete each other locally, so that combining them can achieve a balanced model that does not bias over any subset of data points. For example, model A tends to overestimate the risk of small companies while model B tends to underestimate the risk of the same group of companies. In that case, we may decide to combine the two models to achieve a balance at the level of bias.

Second, *model diversity* is proposed to find interesting variables for partitioning the data points into groups. This help analysts to identify variables that can partition the data spaces into subsets that can generate local models with diversified ”fitness”.

Third, *model representivity* is designed to identify local models that share model coefficients so that the underlying data subsets could be potentially merged and generate a representative model which covers larger area of the overall data space.

## 1.4 Contributions of this Dissertation

The goal of this dissertation is to apply visualization and guidance to the local pattern discovery of identified three tasks while performing data analytics. The contribution in the above areas are summarized as below:

### 1.4.1 Model-driven Data Exploration:

This work reduces cognition overhead of analytic tasks while performing the complex data analytic tasks. It utilizes multi-model abstractions to describe the data in a meaningful and compact way with visual comparison and contrast. 1) A novel data exploration approach is designed by providing plug-and-play multi-models for data reduction. 2) Four linked spaces are offered to support analysis with a connected context across different spaces. For example, data filtering in data space and model comparisons in the model

space. 3) First, evaluation is conducted for this exploration approach via a case study using stock market data. Second, the user study is conducted to compare the alternative view design choices for visualizing the data and model relations. The metrics we use include user performance and user feedback.

### 1.4.2 Model-driven Feature Analysis:

This work facilitates the feature exploration process by combining clustering, ranking and diversifying methods to extract more descriptive features of a high dimensional dataset. It allows analysts to explore the feature space by visualizing their relationships using clustering techniques. 1) **Multi-Stage Analysis:** We offer flexible integration between the automatic processes and user interaction. The analysts can choose the degree of automation and choose to get involved in specific phases of the process. The resulting dimensions can be automatically determined and manually refined at different granularities. 2) **Redundancy Detection:** Our system utilizes both redundancy detection and dimension ranking for the feature selection process. Each feature is clustered into a feature cluster using feature similarity metrics and then ranked by how well it represents the cluster. 3) **Multi-Selection Criterion:** Partition driven analysis as well as multiple metrics are used to identify different sets of features of interest. With visual support in FeaVis, analysts may discover alternative selections of data features that may be interesting to look at.

### 1.4.3 Visual Guided Model Diagnosis:

Enhance the evaluation process of the model quality measurement by providing interactive feedback on how the analytics input affects the performance of models both globally and locally. 1) This work allows analysts to interactively build and evaluate models at both global and local scales. The interactive exploration is guided by the visual designs

in three model spaces. 2) This work utilizes a pairwise comparison of local models for model refinement. Models that complement the *to-be-refined* model are identified and combined *(union of variables)* to the to-be-refined model. 3) This work integrates a novel partitioning strategy for isolating local linear patterns. Strong and weak trends (in terms of goodness of fit) are visualized distinctly in a pattern space. 4) A hierarchical view is presented for grouping local models, where each group can be interactively divided into smaller ones interactively. Meanwhile, the analyst may investigate the relationship between the size of a group and the divergence within it.

## 1.5 Dissertation Outline

The rest of this dissertation is organized as follows:

Chapter 2 discusses the first topic *model-driven data exploration*. We investigate multiple modeling techniques for data abstraction and using the proposed technique to identify local co-movement of time series data.

Chapter 3 presents the second topic *model-driven feature analysis*. It illustrates feature relationships using multiple feature similarity metrics. The partition based similarity analysis are also supported for discovering local feature similarities.

Chapter 4 describes the third topic *local model diagnosis*. It enables analysts to dive into local data space to diagnose how the model performs locally which helps them to refine models locally.

Finally, Chapter 5 summarizes this work and discusses future directions.

# Chapter 2

# Model-driven Data Exploration

First of all, a *model-driven data exploration* framework is presented in this chapter, where data models such as linear model and time series model can be applied to summarize the data space to facilitate the data exploration process for gaining insight. It empowers the analysts to select the predefined methods to summarize the data. This component provides multiple linked analytic spaces for interpretation at different levels of abstractions. For example, the low level data space supports data binning while the high level model space offers model summarizations such as clusters or trends. It also supports model analytics that visualizes the summarized patterns and thus enables the analyst with ease to compare and contrast them. In this dissertation, we provide novel methods for investigating co-movement patterns of timeseries dataset that is important for applications from medical sciences, finance, business to engineering. The models are mainly used as magnifiers analogous to a map reading task. The models automatically capture certain interesting aspects of the underlying data space. In this work, multiple models are utilized to provide a plug-in-and-play style data summarization and interpretation workflow. This work is accepted as the best paper of VDA 2016 [ZWRH16].

## 2.1 Preliminaries of Data Patterns and Models

In this paper, we provide support for co-movement analysis in both the *data space* and the *model space* by offering integrated visual presentation support. Co-movement pattern is a widely studied pattern in application domains, from medical science, finance, business to engineering. It refers to the correlation between a collection of time series objects such as EEG signals recorded from multiple channels or the stock price of different companies.

**Co-movement** *in our work concerns the correlation between time series in both data space and space.* The data space corresponds to the observed values of the time series. Numerous tools have been developed to analyze correlations in data space, such as covariation [KP08] and detrended cross-correlation [RRCZ14]. A derived space is then formed based on the extracted features such as frequency [FGP+13], trend [BPS14], seasonality [CL98], and uncertainty [BTV14] of the time series. The co-movement is a widely studied pattern of time series. The study of EEG co-movement in neuroscience [FGP+13] aims to detect the epileptic seizure onset zone by investigating the causal relationship between different EEG channels in the frequency space. In finance applications, the co-movement research aims to detect financial contagion which is said to indicate the spread of market disturbance [KP08]. The analysis of co-movement patterns in engineering can be used to optimize wireless device localization [CEG+09]. While we focus on financial time series in our work, the proposed framework can be applied to other applications by integrating appropriate domain-specific machine learning techniques.

**Modeling techniques** in this work are mainly used on time series data to detect co-movement patterns by extracting model descriptions. These model descriptions (i.e. trend, seasonality and volatility) are essential for the exploration of the model space in MaVis. A number of techniques have been discussed in different fields for the detection of co-movement patterns. For example, the rule-based approach [WFYL08] designed

co-moving rules to categorize the pairwise relation of two time series as 1) up-up, 2) down-down, 3) up-down, 4) down-up. Unfortunately, these rules create a variable number of segmentation points depending on the dynamics of the time series. For a collection of time series the rule space may thus explode. Analogues to the signal decomposition process (e.g., high vs low frequency) for most signal processing techniques [Mal89], we instead look for statistic models that can describe the co-movement of time series in the model space. In this paper we in particular focus on three common model types for time series, namely, drift, seasonality and volatility. Each of them may be associated with different semantics in the domain.

The models in MaVis are compact descriptions of the raw data such as clusters, trends and others. They are visualized and presented in a derived *model space* to provide a *compact representation* (e.g., cluster radius, slope and etc.) of the original raw data. The cognitive load required by analysts to make sense of the data can be significantly reduced by using machine learning models that lead to very compact descriptions of the data. For example, 1 million data points can be effectively reduced to $k$ clusters ($k \ll 1$ million) in the *cluster model space* so that the analyst can grasp the underlying data space. While there is a need for high performance modern machine learning algorithms, dealing with large scale data is not our primary focus. Our focus instead is related to the second half of the chicken-and-egg dilemma when an analyst may find a pattern not interesting or he/she does not know what is interesting, specifically, we aim to support analysis needed in the following scenarios: *1) what if the extracted clusters are not considered interesting by some analysts? 2) what if the analysts are not sure which models are more interesting than others?*

To tackle the first issue, we list the selected model descriptions that enable the analysts to swiftly examine and determine what model type to examine further. To deal with the second issue, we enable the analysts to engage in the *exploratory data analysis* workflow

of testing multiple methods and comparing them to reach a final conclusion. MaVis incorporates three commonly used models and a higher level analytic space, namely, *model relation space*, to support such comparison activities via linked views. For example, to determine whether linear or non-linear trends are more appropriate to describe the underlying data, an analysts may want to compare the two models in the *model relation space* and decide which model type reveals more interesting patterns.

The model descriptions, however, are dependent not only on the model type but also on the local data partitions that are used for creating models. As discussed in [ZWRH14, MP13], the description of a model (e.g., slope of a trend) is also strongly based on the partitions of the data space. For example, the trend slope of this year's data may be different from that of last year's. To get an overview of the data space, the MaVis *model relation space* thus supports the relationship analysis of the local model descriptions. However, investigating such phenomena clearly adds complexity to the comparison analysis of the *model relation space* as there are, for instance, many ways to partition the space. To facilitate such analysis, MaVis provides analysts the capability of managing and comparing their discoveries in a *nugget space* to keep track of the findings of an analyst so far during her discovering process. A nugget contains a subset of the points of interest produced by summarization for future analysis. For example, when an analyst identifies two clusters in two different data partitions, the *nugget space* maintains summaries of such observations which may lead to other discoveries related to their summaries such as the overlap of two clusters.

Next, we discuss three common types of models for time series data. Each of them is extracted by automated modeling techniques from the literature [PK10, GB14, VBB12, MZ08, Ulr13] which are developed by other researchers.

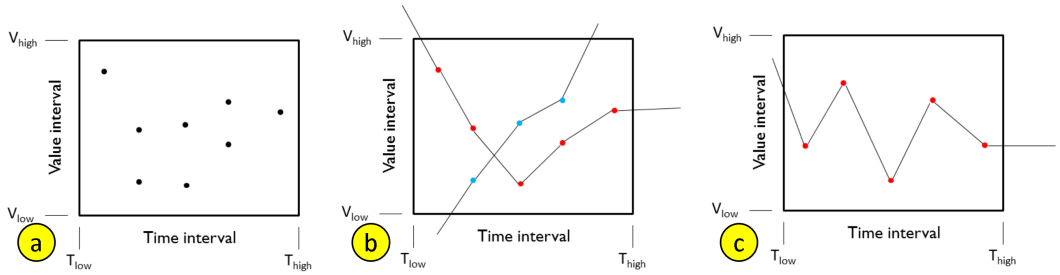


Figure 2.1: Comparison of two binning strategies for collection of time series. The binning method may count every data point (a) or count the number of time series (b). Counting every data point highlights grid cells that have multiple occurrences of data points but with only one time series (c).

### 2.1.1 Drift Model

*Drift model* is often used to describe the increase or decrease tendency of a non-stationary time series. It models the growth or decay of time series data. In finance it is often used as an indication of whether buying or selling a stock is likely going to produce a profit or not. Geometric Brownian motion [PK10] is one of the commonly used techniques to model the drift of financial time series. The Stochastic Differential Equation (SDE):

$$dS_t = \theta S_t d_t + \delta S_t dW_t$$

is often used to simulate the geometric Brownian motion. Many techniques (as summarized in [GB14]) may be used to estimate the parameters in the SDE, including the drift parameter $\theta$. In our work, we integrate the pseudo-likelihood method implemented in R [GB14] into our system to extract the drift from time series data.

### 2.1.2 Seasonal Model

Seasonality may be extracted from time series for prediction and modeling purposes. For example, the sale of ice cream could reach a peak during the summer and a valley in

the winter. Such pattern can be widely found in finance [KKL12], economy, medicine [MUCM03] and other fields. Understanding the cyclic pattern of a collection of time series is informative particularly in the context of co-movement patterns. For time series that move with similar periodic duration, they are more likely driven by the same factors and thus co-move together. Many techniques in different applications have been proposed to investigate such seasonal patterns including wavelet [Mas08], ARIMA [VBB12] and HP Filtering [KM99]. Since we focus on financial applications, we choose to integrate the ARIMA model parameter estimation [MZ08] into our system. The ARIMA model can be used to estimate the most likely cycle duration of the time series. Thus we use it here to represent the degree of co-movement regarding the seasonality duration. Stocks with longer seasonal (e.g. year) duration may co-move with others with similar durations rather than those with shorter durations (e.g. week).

### 2.1.3 Uncertainty Model

Investigating the uncertainty of time series may help us to quantify the degree of risk in finance (stock price data) or help detect brain activities (EEG data). Clearly, different application domains may favor different notions for capturing uncertainty. For example, uncertainty could refer to the volatility of data [Blo09]. It may also refer to the unpredictability of model parameters [BB01]. Also, uncertainty is an interesting problem in data visualization where it refers to errors that occur during the transformation process from data to visual representation [BOL12].

In our work, we focus on the uncertainty of the time series data. In the finance domain, risky assets tend to have certain similarities in terms of their dramatic price changes. In such cases, an investor may gain/lose a lot during a short time period due to the high dispersion of price values. The techniques for modeling such change can be divided into two categories: historical volatility [Ale08] and implied volatility [ABHA09]. Since the

implied volatility is commonly used for risk forecasting, we focus on historical volatility modeling to serve as a volatility descriptor. We adopt and apply the implementation of volatility calculation from [Ulr13] into our system.

We next discuss how to investigate the co-movement in an interactive environment using the above discussed modeling techniques.

## 2.2 Proposed MaVis Framework

In this section, we describe the design and implementation of the proposed MaVis framework designed to support visual explorations in four spaces at different levels of abstractions, namely, *data space*, *model space*, *model relation space* and *nugget space*. The design of the 4 space architecture of the system is based on both the notion of *ladder of abstraction* [Urs13, Vic11] and the idea of *multi-scale representations* [Kin06]. The *ladder of abstraction* illustrates the thinking process that starts with specific items and continues to high levels. For example, the model space (e.g., clusters and trends) provides high level compact descriptions that the analysts may comprehend with ease.

Any given model may not always be perfect in terms of conveying accurate and useful insights. It is often unclear how well a given model describes the original data [CWRY06] due to the fact that there can be information distortions during the data abstraction process from data to visual representations. One type of information loss during the abstraction process is due to the existence of local patterns that cannot be described by the global pattern [ZWRH14]. We use a multi-scale representation strategy to model data at multiple granularities so that local patterns of interest are no longer lost. In order to support multiple granularities, MaVis provides user controlled scales for capturing local patterns. These local patterns, once detected, are then presented in a *small multiples* display to the analysts. Then, the local patterns and the global patterns may be compared and contrasted

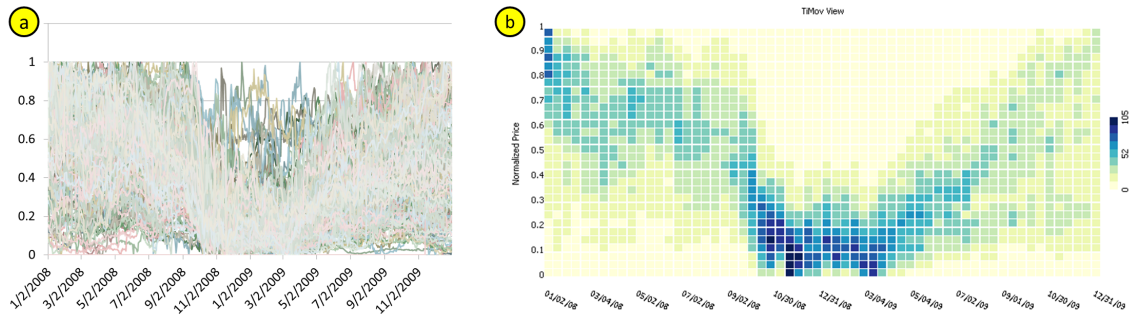


Figure 2.2: *Time line movement* view (b) presents a collection of 250 time series where x-axis represents the time progression and y-axis is the normalized price values ranging from 0 to 1. The darker region in the view at around October 2008 shows that the majority of the companies were at relatively low price values. The line chart view (a) presents the data with the same normalization method (view rendered within Excel).

via the designed *linking operator*. Next we discuss in detail the design and implementation of the 4 spaces.

### 2.2.1 Data Space

The data space of MaVis supports data specific analytic queries (e.g., brushing over a period of time) that allows the analyst to investigate the co-movement of time series at specified time intervals. One common approach for visualizing the data space is to map the time series to segments of lines in a line chart (Fig 2.2a) (similar approach can be seen in [HS04]). Its variations such as the ThemeRiver based designs [SCL+12] are also popular in cases when a moderate amount of time series are displayed. In MaVis, we seek for an alternative visual representation that is inspired by the idea of binning aggregation [LJH13]. The binning strategy provides an overview of all the data before the analyst submits any queries. The line chart approach tends to work well when one wishes to examine a detailed view of a collection of focused time series but the view may be overwhelming at first glance due to the high density of time lines [HS04]. To overcome the clutter of the line chart view we design a *time line movement view* (as

shown in Fig 2.2a). The view illustrates the movement of a collection of time series at a relative (i.e., percentage) scale. The absolute scale may reveal other patterns, however, we choose to use relative scale as the degree of growth in finance is often measured by percentages.

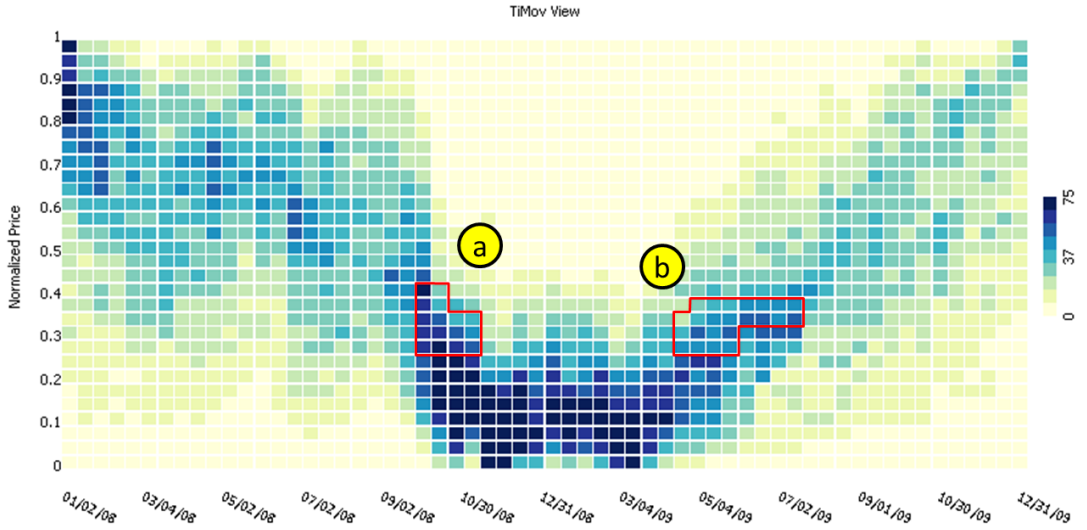


Figure 2.3: Two constraint boxes are placed to reveal companies that fell (a) and rose (b) during the 2008 crisis. Compared to the view in Fig 2.2b, we see that most ( 70/ 100) of the prices move with such behavior. The color schema range is adjusted based on the maximal count of all the grid cells by default.

The *time line movement view* as presented in Figure 2.1 transforms the collection of time series into a value-time space. Color is used to indicate the population densities within each grid cell. Darker color indicates higher density while lighter shows lower density. The horizontal and vertical scales are adjustable and controlled by the user depending on their needs. To observe sensitive value changes the user may adjust the vertical scale to finer resolution. Similarly, to perceive short term pattern changes the horizontal scale may be adjusted. The idea of adjustable bin is motivated by the design mantra *"Overview First, Zoom and Filter, Details-on-Demand"* by Ben Shneiderman [Shn96]. By adjusting the bin size, the user can filter time lines at a controlled resolution and observe the co-movement pattern in detail.

Next we discuss the two options we considered for the binning method. The first option for binning the time lines in the *time line movement view* is to count the number of values that fall into each grid cell (Figure 2.1a). This method is memory efficient regardless of the size of the dataset. It only requires one scan of the dataset and then to count the number of data points in each bin. The memory requirement is determined by the resolution of the *time line movement* view. However, it is dependent on the sampling rate of the time series (i.e., hours, days or weeks) which may distort the view. The second option is to count the time series (Figure 2.1b) that go through each grid cell. The purpose of only counting the number of time lines is to reduce the impact of variances within each grid cell and highlight the overall pattern for a collection of trajectories (Figure 2.1c). It requires extra memory to store the index of the time lines so that we remove all duplicated data samples of each time line within a particular grid cell.

To further support the exploration in the data space, two interactive operators are integrated into the *time line movement* view of MaVis, namely, filter and link. The filter operators allow the analysts to apply constraint boxes similar to those in [HS04] at the resolution level specified by the analyst via adjusting the size of the bins. We consider two options for designing the filtering operator: *preserve* and *exclude*. That is, the behavior of a filter selection is either to preserve the items that are selected by a user or to conceal them. To facilitate the refinement of filtering, we support multiple selections which are aggregated with set operators such as *union*, *intersect* and *negation*. With the filter aggregation, the selection query box is more flexible than a typical single rectangle box. For example, an analyst may want to exclude some the time series from those that bypass a large rectangle. For this, she may attach a small *negation* rectangle to the larger box (as shown in Fig 2.3).

The *linking operator* links the analyst selection in the data space to model descriptions in the model space to enable the analyst to further examine the co-movement of the

selected time series regarding other domain specific features such as drift (for stock price analysis).

Additionally, to support multiple resolutions of the binned view. The size of the bins is adjustable by the user in two directions, namely the time axis and the value axis.

### 2.2.2 Model Space

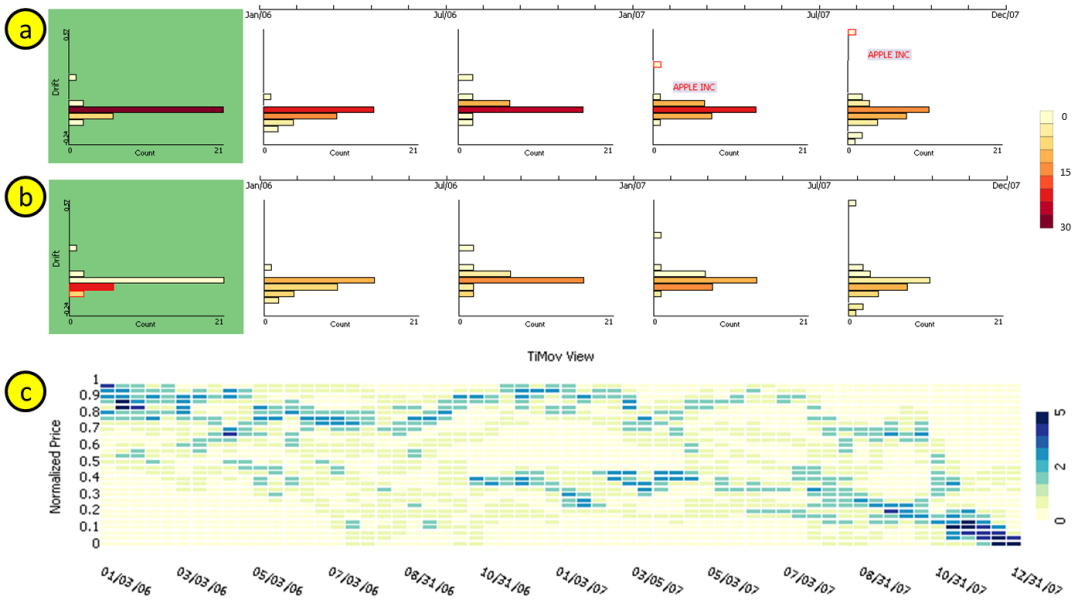


Figure 2.4: Drift abstraction of a collection of 32 time series objects. a) The default color encoding which represent the count of time series in each bin. b) Filter operator selects time series lower than the risk neutral zone. The color encoding represents the count of selected time series. c) Link the selected time series in space b back to original data space. The leftmost histogram shows the overall drift of the time series over the selected time span (2006 and 2007). The histograms to the right with white background show the local drift of each company at the granularity of 6 months in each view. In these sets of views, we observe several interesting patterns. (1) Most companies stay in the risk neutral zone which is the longest bar in all the histograms while many companies fell down at the end of 2007. (2) We can also observe an outlier time series (Apple) that grows exceptionally. (3) Linking from the model space view (highlighted rectangles in leftmost rectangle of b) to the time line movement view reveals an overall falling pattern with high density towards the end of 2007 in (c).

In this section we focus on the three models we discussed in Sec 2.1 for time series

data modeling, namely, drift, seasonality and uncertainty. The drift indicates whether buying an asset yields potential profit. The seasonality represents how predictable the change of a stock price is. The uncertainty (also called volatility) of a stock price measures how much the price may change over a certain period of time. The above modeling method may generate a description that explains certain domain patterns. For example, let us take a closer look at the stock price of a particular company: *Apple, Inc* (Fig 2.4a). The overall drift of Apple is 0.35 in the years of 2006 and 2007. This is a indication of a relatively strong growth. The finer resolution reveals local dynamics that contain more information. In this case, the drift of Apple is 0.29 in the first half of 2007 and 0.57 in the second half. This means the growth of Apple in the two years mainly concentrated in the second half of 2007.

One interesting question to answer is which companies have similar drift patterns like Apple or any other company of interest? We design the *model similarity view* (Fig 2.4a&b) that visualizes the similarity of time series in the model space. Next we discuss how the model space works as well as how the visual representations are designed to illustrate the local dynamics.

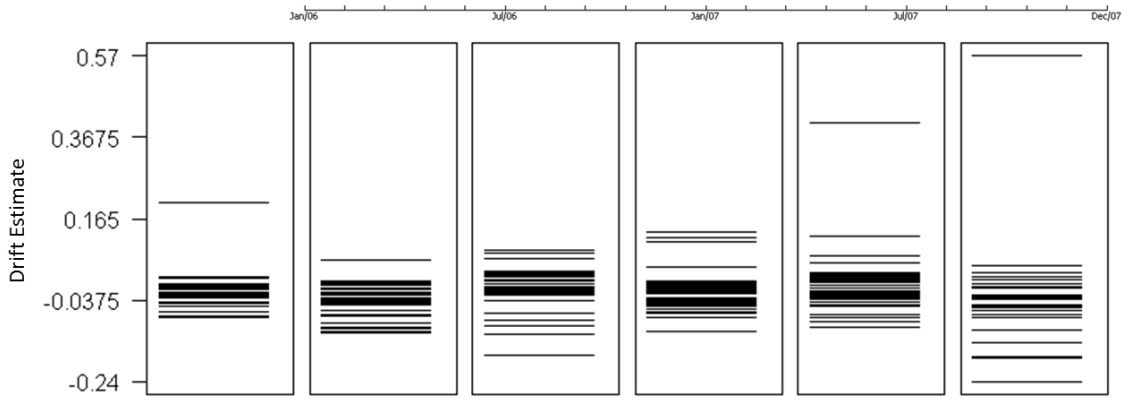


Figure 2.5: Time series similarity in the drift model space. The leftmost *bar code view* visualizes the overall drift tendency of the selected time series where each line corresponds to one time line. The 5 bars to its right visualize the local drift.

The model space of MaVis provides an abstracted representation of the original time

series data to highlight any domain related co-movement patterns such as correlation between price risk of different companies. The domain related co-movement patterns are revealed by utilizing the abstracted description of domain models such as *Brownian motion* (drift abstraction) and *Weighted moving average* (volatility abstraction). Compared to the automatic *piecewise linear approximation* method [KCHP93], our primary objective is to facilitate the sense making of the analytical process rather than finding the best data points to preserve for further analysis. Therefore, we use both the domain specific modeling techniques (discussed in Section 2.1) and a user controlled interactive segmentation for extracting local patterns at specified time interval size.

We chose the user driven approach due to several reasons. 1) The automatic segmentation points extracting methods tend to work on univariate time series. They are not appropriate for a collection of time series because finding the alignment of segmentation points for a collection of time series is not a trivial problem. 2) Manual segmentation would be controlled by the analyst. The analyst thus may choose a universal cutting point for the collection of time series based on the overview of the data space. For example, the crash of the stock market in 2008 lasted about 6 months before recovering when we look at the *time line movement view* (Fig 2.2b). The analyst may thus choose to select the 6-month resolution as a reasonable setting to explore the local *model space*.

To present the co-movement of time series in the model space, we consider several options. 1) Present the model estimate (e.g., drift) of each time series into a 2-D projection where one axis represents the estimated value and the other axis represents the order of the data points. However, we face the dilemma of optimizing the ordering of data points across different projections and preserving the group structure of similar model estimates in the same time. 2) To optimize the presentation we instead turn to a 1-D layout (*bar code view*) that only shows the value of model estimate (Fig 2.5). Each line segment of equal length represents the drift of a corresponding time series. The vertical position of it

is determined by the estimated drift value. With support of brushing and linking, the bar code view is able to illustrate the co-movement pattern represented by connecting the line segments.

However, the line connections may be difficult to interpret when line segments overlap in several regions. It is especially difficult to interpret when the density of line segments is high.

To overcome the above clutter issue we use a histogram view (Fig 2.4a) by binning the line segments. The length of each histogram bar represents the count of line segments. The color encoding is used to represent the number of line segments that are currently highlighted (darker color means higher density of line segments in that bin). For example, when an analyst applies a filter operation to select the bins that represent time series with low drift estimate in the 2 year view (leftmost in Fig 2.4b), the color of all bars is updated accordingly to show the prevalence of the selection in other bins. It represents how these time series are distributed over the 4 local views (e.g., the first half of 2006). The design for model space visualization is evaluated in our user study described in Section 2.3.2.

There are two types of brushing and linking operators in the model space. The first type is the linkage between multiple model space. The co-movement pattern in one model space can be linked to another model space. Such linkage may reveal relationships between different model types or across multiple time intervals. Understanding the model relationship may help answer several questions including: *What are the volatilities of a selection of growing time series?* or *How does the drift of a collection of time series change over time?* We will discuss the design for analyzing the model relationships in detail in Section 2.2.3. The second type is the linking between the model and the data space. Specifically, the patterns in the model space can be linked back to the data space to reveal the data characteristics. For example, by selecting the time series with a low drift estimate in the drift model space (Fig 2.4b), the overal time line movement pattern

is shown in the data space (Fig 2.4c).

### 2.2.3 Model Relation Space

The primary purpose of model relation space is facilitate the investigation of the co-movement dynamics. The hypothesis of a co-movement pattern within one model space during one specific time interval may be reinforced or lessened in another model space over the same or a different time interval. For example, even when two companies have a similar tendency of growth (i.e., drift), the degree of fluctuation (i.e., volatility) can differ greatly. Therefore the co-movement pattern we observe regarding a single model type may be biased. On the other hand, the growth tendency may also diverge over time. It may indicate that the co-movement pattern only occurs within a specific time interval. To capture such dynamics and to compare multiple models we visualize each model type in one row of an integrated small multiple display. The analysts then can compare and contrast the patterns interactively.

We use a similarity metric and color encoding to illustrate the pattern overlap of multiple models. To measure the degree of overlap, we first apply the *Jaccard similarity* measure between the focused model space and non-focused space. In a focused space, the analysts brush and select time series of interest. In a non-focused space, each bin of time series are grouped by co-movement properties (e.g., similar drift). When we are interested in whether a selection of 20 time series in space A are still co-moving in space B. We can check if any bins in space B contain every time series of the selection. We choose to use *Jaccard Similarity* as it is a commonly used measure for set similarities:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

where $A$ and $B$ are two sets of time series.

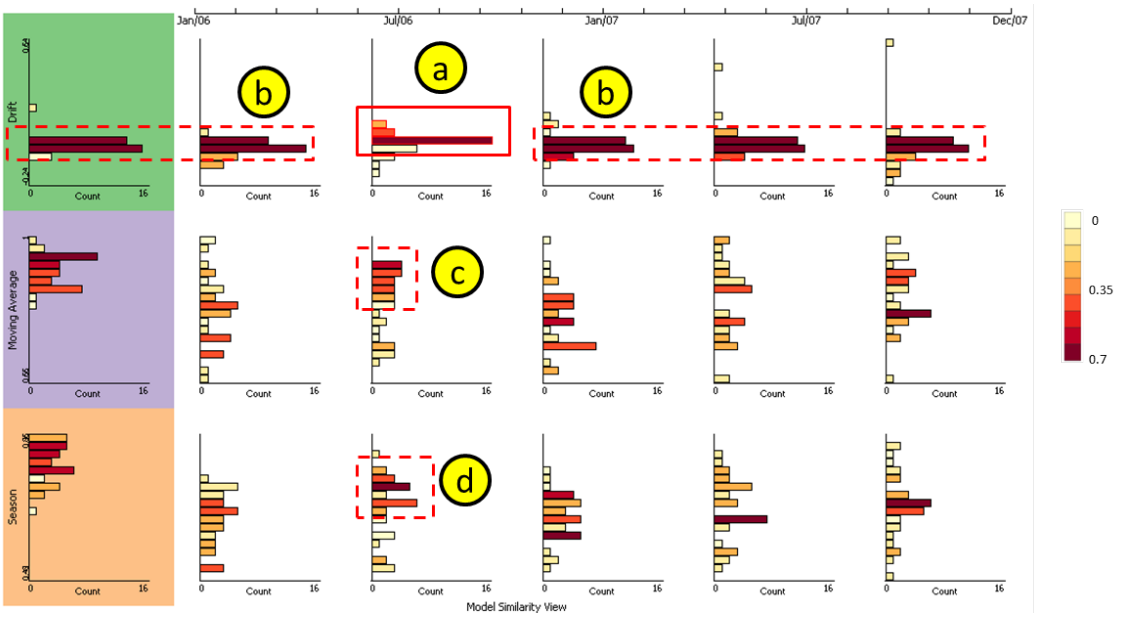


Figure 2.6: Model similarity analysis view. a) A brushed co-moving drift pattern begins since about July 2006. b) The darker color bins show a high correlation between different time intervals. The drift estimate of bins in (a) and that in (b) are at relatively the same value range. It shows the drift of co-moving patterns is quite consistent over time. c) A high degree of volatility is shown. d) A long seasonal cycle is represented.

After computing the similarity, we update the color of bins (Fig 2.6) to represent it. In the case when multiple bins are selected (e.g. 3 bins of time series are selected in Fig 2.6a), we use the union of all the selected bins as set $A$ and the other bins (e.g., bins in b, c and d) as set $B$ to compute the similarity.

### 2.2.4 Nugget Space

The design of the nugget space is to support the analysis of multiple user queries in one place. A nugget is a subset of data points selected by an analyst in a user query via brushing or filtering. For example, it can be created when an analyst brushes over a set of time series in one model space based on how closely they are related. In this space, we are particularly interested in how the co-movement patterns are different over time and under the models of different types. A pattern is defined by a user query over a particular time

interval and the pattern difference is measured by the similarity between those queries. The objective of this analytic space is to answer questions such as: *1) How closely does the current high risk (i.e., high volatility) relate to an increasing trend (i.e., high drift) in a later time? 2) How many time series are present in such a pattern?* To answer these questions, we provide two features: 1) Summarize the user queries (e.g., risk vs. growth) and then 2) compare them to establish connections. In the *nugget space* we support the above two features by visualizing the summary information in a *nugget analytic view* (Fig 2.7) where the queries are compared and analyzed.

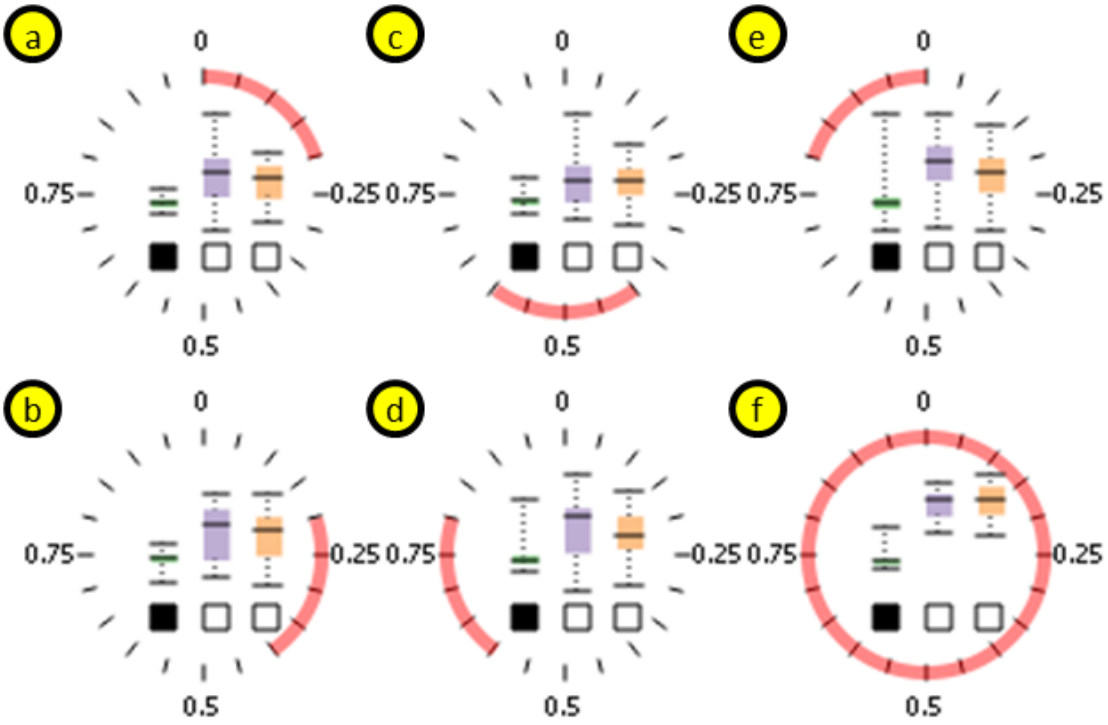


Figure 2.7: The view represents a collection of time series with a co-moving trend that is identified in the first time interval indicated by the green box plot (a). However, the co-movement pattern of the same group became gradually diverging over time and reached peak during the last time interval (greatest variance indicated by the height of bars) (e). From a long term perspective, the co-movement pattern that is identified in the green model space is more consistent across the three model types at time interval (f) compared to the other local intervals (a-e).

**Nugget summarization:** First, we describe how to summarize and visualize a nugget

that is created by a user query. For each nugget we present three types of information: the *time interval* of the user query, time series *distribution* for each model type, and the *model type* which analysts choose to model the time series and submit a query. Inspired by the clockmap view [FFM12], we use a round shaped glyph to present the summarization information (Fig 2.7). 1) The outer space of the glyph is reserved to display the *time interval* of the user query. 2) The time series distribution of each model type is represented by 5-number summary, namely, min, max and 3 quartiles of the corresponding model description of the selection of time series. 3) The inner space of the glyph displays the distribution of model descriptions of one of the three model types. To further explain our design, the *Box-and-Whisker plots* for the distribution are color coded to match each model type. A small rectangle underneath each box plot is used to indicate the model type of the user query (analogus to a tickbox). The three box plots in each glyph describe the distribution of all three model types for the user query that may lead to insights about the data. For example, in Fig 2.7c, the drift pattern (green box plot) shows the selected time series are co-moving with a rather small dispersion, yet the volatility measure is quite diverging as the height of the second bar (volatility) is relatively high. It suggests that determining co-movement of the selected time series only by the drift is biased.

To determine the way of visualizing the summarization, we have experimented with several glyph design alternatives. We then finalized our design based on user feedback. For example, the time interval can either be represented in a circular (i.e., 360 degree) space or a linear space. We choose circular space because degrees in the circular space can support the comparison of angular values between two glyphs without alignment as we believe degrees are more interpretable. We also hypothesize that it is more challenging to perceive the time ordering of any two glyphs in a linear space unless they are properly aligned (evaluated in Sec 2.3). We also experiment with the visual designs for indicating model types.

**Nugget comparison:** A second feature of the *nugget analytic view* is to provide comparisons between multiple nuggets which covers different data subsets. There are several ways to quantify the similarity between multiple data subsets. One way is to compare the data sample distributions to see whether they are from the same one. However, there is no readily made solution for time series collection as even for one single time series, the distribution may change over time. Then a plausible alternative approach is to make use of the already computed model description for each time series. We use the query overlap measure and the query summarization together to compare the similarities of user queries. Specifically, to compute the summary of a given pattern, we first convert the 5-number summaries to a vector of length 15 that consists of 5 values for each of the 3 model types, respectively. Let $\mathbf{v}_a$ and $\mathbf{v}_b$ be the vector representation of two patterns $A$ and $B$. The similarity score is computed as:

$$s(a,b) = \frac{|A \cap B|}{|A \cup B|} * \arctan\left(\sqrt{||\mathbf{v}_a||^2 + ||\mathbf{v}_b||^2 - 2\mathbf{v}_a \cdot \mathbf{v}_b}\right)$$

The similarity measure above is a combination of pattern overlap measure (*Jaccard similarity coefficient*) and pattern summarization measure (*Euclidean distance*) normalized to [0,1] space. Since the similarity is a pairwise relationship, another problem we need to solve is to display the $n$ by $n$ similarity relationship in addition to the $n$ glyphs already displayed which is likely overwhelming. Thus, we design a color filter on the alpha channel of the color space to fade the glyphs depending on how similar they are to the focused one so that similar nuggets can be recognized (Fig 2.9 second row). The similarity score $s(a,b)$ between two nuggets (a and b where a is the highlighted glyph at bottom right corner) is also displayed on the top left corner of each glyph.

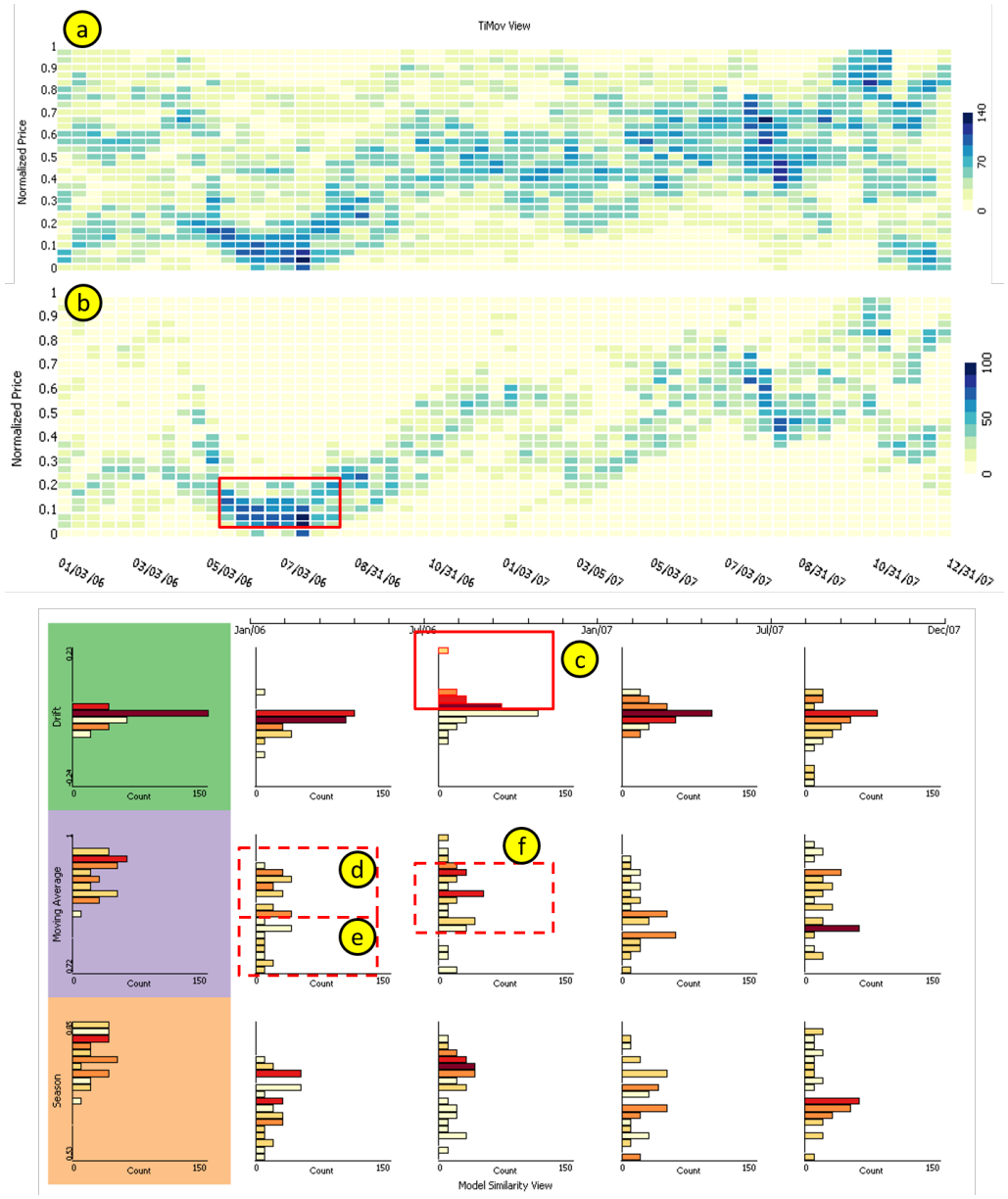


Figure 2.8: The views show a interactive exploration process for co-movement pattern investigation. a) The overall drift pattern is presented as heatmap view. b) Filtered results are shown after a range query is submitted. In the view to the right, co-moving patterns are linked via color encoding. c) When the collection of growing time series are selected the corresponding risk of this collection is linked to other portion of the views such as (d) (e) and (f). d) The boxes have darker colors which indicates higher correlation. e) The lighter color there shows lower correlation. f) The pattern is also showing some degree of correlation but at high dispersion which means the collection is less likely co-moving.

## 2.3 Evaluation

In this section, we discuss the evaluation of the MaVis framework using both a case study and a user study. The main purpose of the case study is to show the typical analytic workflow of MaVis using a financial stock price dataset. The user study is conducted for testing our system regarding the usefulness and design choices.

### 2.3.1 Case Study: Stock Price Co-movement

The purpose of the case study is to show that MaVis is able to support the discovery of patterns that are interesting to analysts, specifically people who often analyze stock price data. To conduct the case study we collect data from http://www.crsp.com which is a research center for security prices. The daily stock exchange data for all listed companies dates back to the year of 1925 in NYSE and 1972 for NASDAQ. For the purpose of evaluating our system, we collected a subset of the database by querying one category of all the industries, namely, the USA based information technology companies classified by SIC (*Standard Industrial Classification*) code with the range from 7371 to 7379. We also clean the data based on the availability of data points from year 2006 to 2009. Time series with missing values are discarded. After this cleaning process, out final collection contains 348 companies and a total of 348,696 data points.

An analyst may have various questions she wishes to ask of her data before starting the analysis. For example, *"What are the overall co-moving patterns in the data space?"* To analyze the co-movement patterns, the analyst first studies the *time line movement* view (Fig 2.8a) to explore the data space. From the view, she perceives a dominant price fall pattern around *Jan. 2006 - June 2006*. She then has a second question. *"Does the selection of companies comove in the other months?"* She then submits a constraint query to preserve only the time series presenting a falling pattern before and near June

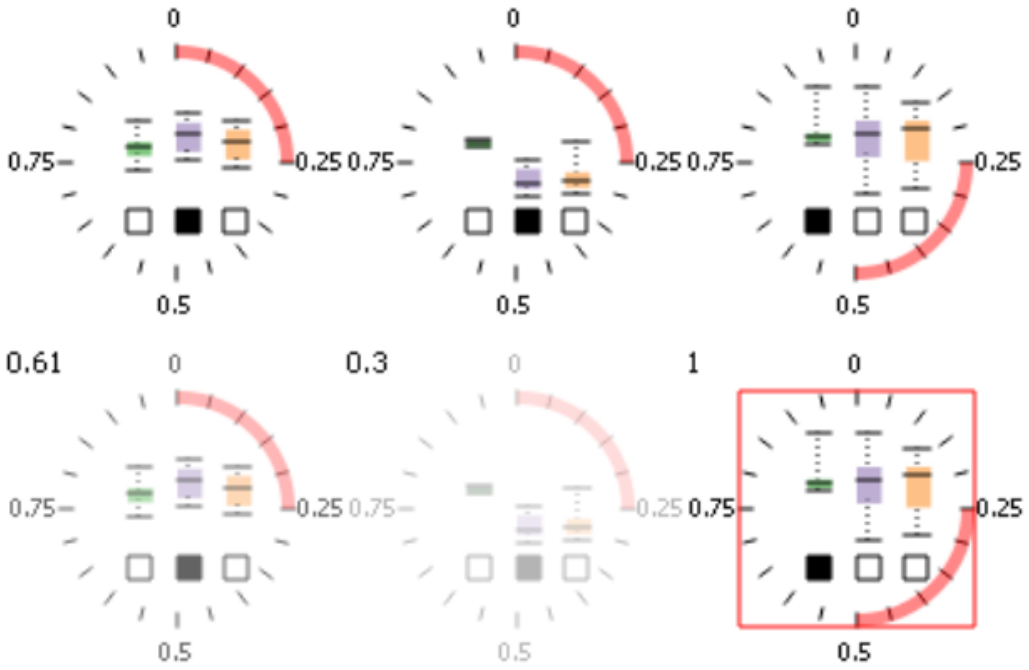


Figure 2.9: The first row (from left to right) shows the summary statistics of the selections in Fig 2.8d), e) and c). The second row shows the same glyphs with focus on a reference glyph for comparison. The similarity score is calculated between the reference glyph and the other glyphs (second row) and then the similarity score is rendered as alpha value of the glyph color.

2006 (Fig 2.8b). After filtering, other perceivable patterns are revealed: the time series start to climb and reach the first high point towards the end of 2006. Later on, starting from early 2007, the time series start to rise again till the end of 2007. On the other hand, the selected collection of time series have an overall increasing trend in the data space according to the view (Fig 2.8b).

After seeing an overall pattern, the analyst may still want to know more details about the dataset. For example, *what are the other characteristics of the falling patterns in June 2006? Are there any fluctuations within the co-moving collection of time series? What are the risks associated with the increasing or decreasing drift tendency?* To get answers to these questions, the analyst moves on to the *model similarity view* (Fig 2.8 right) to study model descriptions for the selected collection of time series. In Fig 2.8c,

the solid line rectangle highlights the user selected time series that have a relatively higher drift estimate among the population during *July 2006 - Dec. 2006*. Then she notices the degree of fluctuations in two time intervals (measured by moving average and marked by dash line rectangles in Fig 2.8d, f) are correlated with the drift patterns. Specifically, the color encoding suggests that the high growth pattern among the population during *July 2006 - Dec. 2006* is correlated with the high degree of fluctuations (i.e., high risks) in *Jan. 2006 - June 2006*. Also, the degree of fluctuations decreases while the collection of time series are growing in *July 2006 - Dec. 2006*. This may indicate that the potentially earning stock time series present high risks before they actually start to earn.

Next, the analyst may still have questions about the co-movement pattern relationship. For instance, she wants to know *how closely are the patterns related*. The color encoding helps her to identify a region of interest and to get an overall sense of where to look next. To further analyze the dataset, she moves on to the *nugget analytic view* (Fig 2.9). The glyph representation of the view is generated by summarizing the patterns browsed by the user. She clicks on the rightmost glyph on the first row which represents the high drift pattern. The second row of Fig 2.9 is used to display the correlation between the selected glyph and the other two. In this case, the analyst found the growth in *July 2006 - Dec. 2006* is more correlated to the high fluctuation co-moving collection in *Jan. 2006 - June 2006* (with a similarity score of 0.61) than the low fluctuation collection in the same time interval (with a similarity score of 0.3).

To conclude the case study, we have shown that analysts was able to uncover an overall market down movement pattern in the dataset. She drilled down and found the fall of the market followed by a growth of most of the companies. Furthermore, the growth towards the end of the time frame is positively correlated to the degree of fluctuations at an earlier time.

### 2.3.2 User Study Design

We recruited 21 subjects including professors and students from the departments of Mathematics, Computer Sciences, and School of Business at WPI. The main purpose of this user study is to validate the *usefulness* and *design* of the MaVis framework. 1) The usefulness test verifies if MaVis is useful to an analyst for undertaking a particular task. It is evaluated by testing whether the useful information is delivered as expected. 2) The design test quantifies how a user interacts with a view compared to other plausible alternative choices. It is evaluated by asking the subjects to answer the same question after looking at either design X or Y. We record the time and accuracy of a subject on both design X and Y. Then we ask for their preferences between X and Y. We randomly swap the order of design X and Y for different subjects to avoid learning effect. The accuracy is measured by which percentage of the subjects can get the right answer. The design X is the chosen design in our system.

Next, we describe the user study design in detail. We ask each subject 9 questions about the design 3 views in MaVis (3 question per view). The expected time to finish is about 15 to 20 minutes based a pilot study involving a small sample of 3 subjects (not included in the 21 subjects). The 3 questions for different views are in a similar format. The first question (A) asks the subject to determine if she/he can spot a specific pattern in either design X or design Y. The second question (B) asks if the subject has more questions he/she wants to ask the system as follow-up questions. The third question (C) asks which design a subject prefers, X or Y.

The visualization of MaVis mainly consists of 3 views, namely, the (1) *time line movement view*, (2) *model similarity view*, and (3) *nugget analytic view*. We label our 9 questions using both the view number and the question number. For example, for the *time line movement view*, we have the following 3 questions:

**1A** Do you think there is a growing pattern that involves at least 100 companies in the year 2007?

**1B** Which of the following question would you like to ask? Choose the most important one in your opinion. 1) How closely are the companies of the growing pattern related in a different time interval? Answering this question may help analysts to understand whether the co-movement pattern in 2007 is consistent over time. 2) What are the names of these companies? Answering this question may help the analyst to confirm the pattern based on their prior knowledge about these companies. 3) Do these companies have other similar properties other than the drift pattern? Answering this question may help analysts to get a broader picture about these companies such as understanding the volatilities and seasonal patterns. 4) Don't know. 5) Other.

**1C** Which design of the two in question *1A* do you prefer?

The choices for any questions are typically like the following. For question *1A*, the user may choose to answer *Yes*, *No* or *Don't know*. We further ask the user to mark the interesting pattern (lines, bars or glyphs) if they answer *Yes*. Only the subject that answered *Yes* and correctly marked the pattern of interest are considered a positive example for the numerator of the accuracy computation. Furthermore, they need to answer the question twice by looking at both design X and Y to validate our choice.

For question *1B*, we want to understand if any further questions inspired by the current view can be answered by the system next. Option (5) is used as a flexible response to capture other thoughts from the subjects. The option (4) is for the subjects who have no more questions and they don't know any other questions next might be interesting. The options (1) to (3) are the questions that can be answered by the system. For example, the question *"How closely are the companies of the growing pattern related in a different time interval?"* can be answered by exploring the model similarity view.

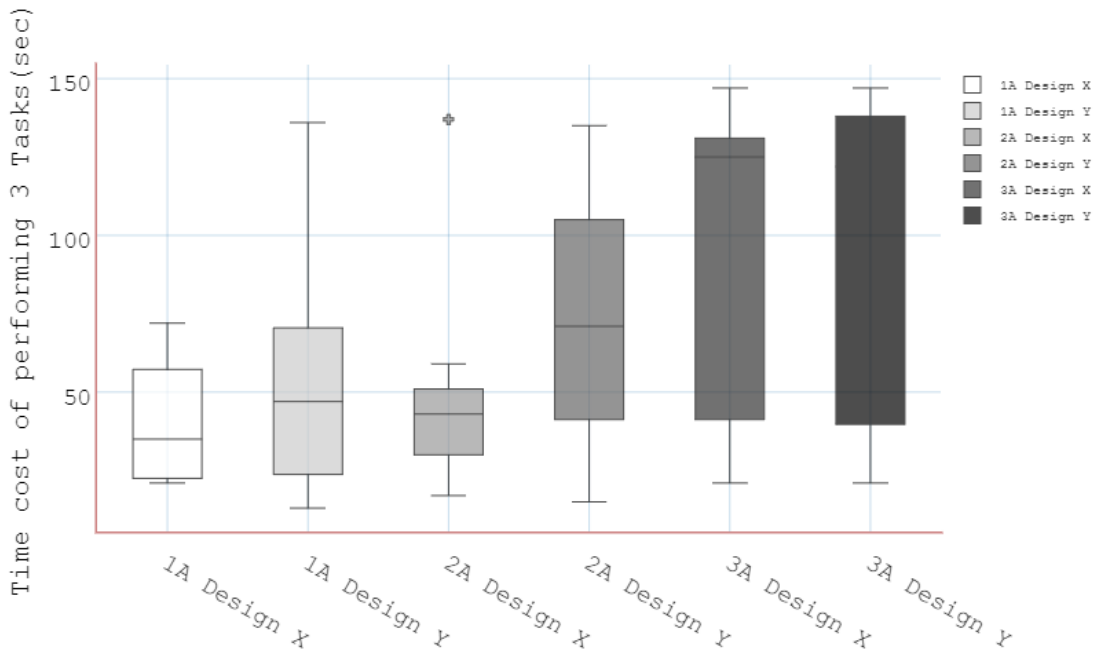


Figure 2.10: The chosen design of the views in question 1A and question 2A requires less time for discovering the pattern of interest. The two glyph views tested in question 3A require relatively the same amount of time. However, the chosen design has better accuracy as discussed in Sec 2.3.3.

For question *1C*, we want to verify our design choices by learning about the preferences of each subject. For example, in question *1A* design X and Y are used. Specifically, based on the literature [AMST11] for multivariate time series visualization techniques, line charts are the most appropriate design to compare with our binned design. As it appears to have the highest information density compared to the other techniques such as ThemeRiver [HHN00], Braided Graph [JME10] and Circle view [KSS04]. We determine our preference based on the time and accuracy measure of these alternative techniques.

The questions for the other two views are similar in style. We discuss the result in Sec 2.3.3. The other 6 questions are designed to evaluate the *model similarity* view and the *nugget analytic* view. The two design choices for the *model similarity view* are discussed in Sec 2.2.2 (barcode view vs. histogram). The two choices for the *nugget analytic view* are discussed in Sec 2.2.4 (linear space vs. circular space).

### 2.3.3 User Study Result

The result of the user study shows that our system is reasonably useful when the subjects are answering the assigned questions. For question *A* of all the three views, the time spent of two groups of subjects for both design X and Y are summarized in Fig 2.10. It shows the time spent on design X (our choice) and Y (alternative) over the 3 type A questions. According to the result the choice we made for both the *time line movement view* (1A) and *model similarity view* (2A) are better (with p-values as: $p_1 = 0.09$ and $p_2 = 0.01$) in terms of time efficiency. We also observe our chosen designs are better in terms of accuracy (Fig 2.12): [0.77 vs. 0.46] for *time line movement view* (1A) , [0.85 vs. 0.15] for *model similarity view* (2A). For the two designs of *nugget analytic view* (3A), the difference is not as significant in terms of time efficiency. Both glyph designs require similar effort to understand. Regarding the view accuracy, the result is [0.54 vs. 0.31] for *nugget analytic view* (3A) which shows our choices are better in terms of accuracy. The p-values are calculated using R package *t.test* [R C12] with option of *two.sided* and default confidence interval of *0.95*.

For question *B*, we count the number of subjects who chose to ask questions that are supported by our framework (option 1 to 3). We also count the number of subjects who have no further questions (option 4). There are also a few subjects asked in-depth questions that are not supported yet (option 5). We show the result of question B in Fig 2.11. According to the result, one user chose *Other* for question 1B (*time line movement view*) and a second user chose *Other* for all the three views. They both left comments about what other questions might be more interesting. These are in-depth questions such as "why do all the companies drop at the same time?". To answer these questions, analysts may need to more work and collect more related data to gain a full picture. Using the dataset we collected is not yet sufficient to answer it. It is beyond the scope of our toolkit. Most of the subjects selected questions that can be answered by the system. It shows that

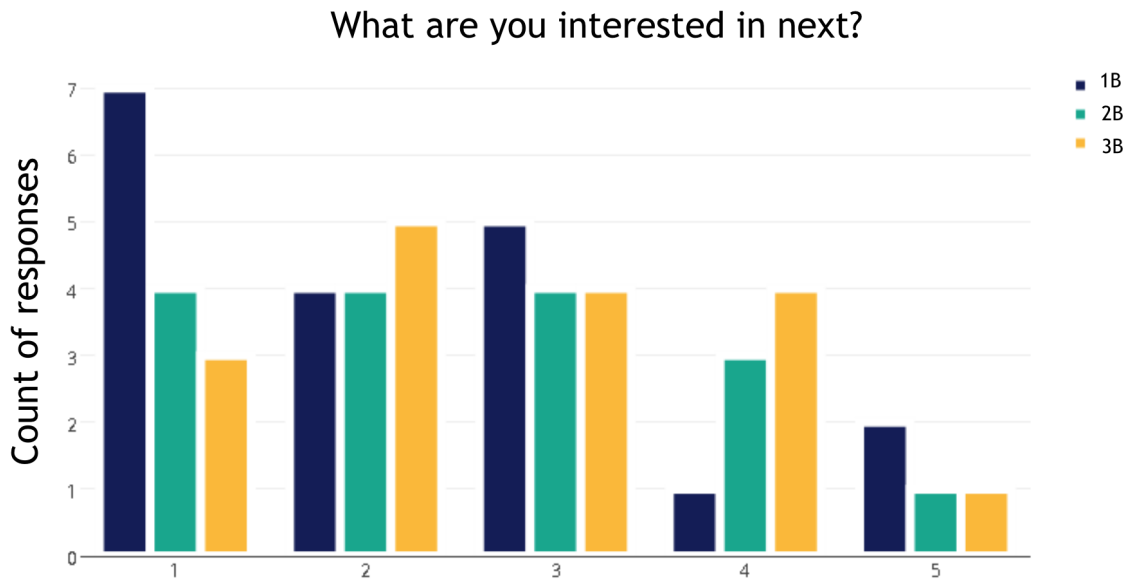


Figure 2.11: Each question B has 5 options (x axis) a subject may choose from. Option 1 to 3 (Sec 2.3.2) for question B are supported by our system and the subject may dig further to discover more insights. Option 4 is *Don't know* which means the subject has no more questions. Option 5 is *Other* and the subject may have additional questions to query the system but we do not yet support those. Bars with 3 different colors represent three views we are evaluating (1B:time line movement view, 2B:model similarity view, 3B:nugget analytic view). Y axis represent number of subjects who chose the corresponding option. Based on the result, few subjects chose option 5 indicating the framework covers most their further needs initiated from the given 3 questions.

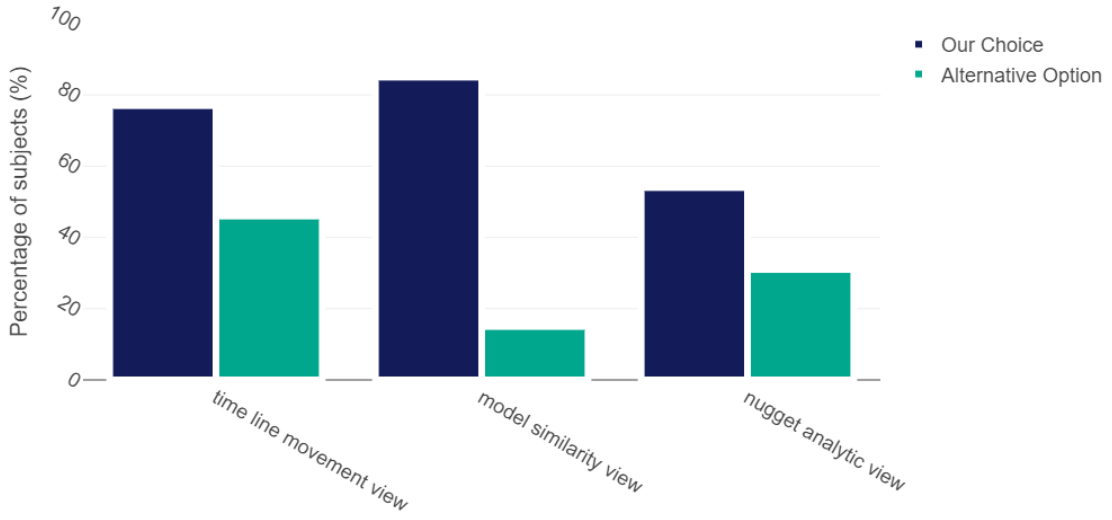


Figure 2.12: Accuracy comparison between our choices and alternative options. Y axis shows the percentage of subjects who correctly recognized the pattern in the design space. X axis lists the design choices we have for the three views.

our system works as expected and it is able to guide the user to further investigate patterns of interest during the exploration process. More subjects tend to choose option 4 in *nugget analytic view*. As we can see in Fig 2.11, the green bar (*model similarity view*) is higher and the orange bar (*nugget analytic view*) is the highest. This indicates that higher level spaces tend to require more effort to interpret.

Task *C* collects the user preferences about the view choices. According to the responses, the percentage of subjects who prefers our final choice are 77%, 92% and 69%. It confirms that we made reasonable choices for our final design.

## 2.4 Related Work

Recently, several works have attempted to utilize model-driven visualizations to help analyzing data. The model-driven approach by Garg et. al. [GNRM08] described a visual

analytics infrastructure that adopts logic reasoning to help reduce the complexity of visual analysis by automating the selection of interesting patterns. This approach has a similar goal to ours in that it aims to reduce visual complexity using algorithmic methods. MaVis provides multiple automated modeling methods for data reduction and additionally allows comparison and contrast between these methods to gain more insights.

Dis-Function [BLBC12] presents a system to learn the distance between data objects with both user input and predefined metrics. It handles low-level optimizations such as distance computation and presents high-level patterns to the user to aid the optimization. In MaVis, instead of learning a single distance function, we aim to support analysts to identify the relationships of time series in multiple model spaces with different ways of measuring similarity. The Nugget Browser [GWR11] displays visual abstractions over data points using clustering techniques which enables high level sub-group pattern discovery. The multiple level abstraction is similar to our approach. In addition to that, MaVis also supports user query analysis in the nugget space to help analyze the correlations between the user identified nuggets.

In many cases, a single learning algorithm or a single view may fail to capture the true characteristics of a dataset. The EnsembleMatrix [TLKT09] designed visual representations to present results from multiple models. The idea of combining different models is similar to our approach. However, their views are designed to support the model assembly process. MaVis is instead designed for data exploration while using modeling techniques for data reduction. Potter et. al. [PWB$^+$09] proposed the Ensemble-Vis framework that consists of a collection of views at multiple scales which inspired our work. It combines views to present information of different types to facilitate the exploration process. The authors of CVVs [JE12] explored visual design spaces for presenting correlated visual representations in case of complex heterogeneous data. These two works focus on coordinating multiple views for complex information visualization. In MaVis, we provide

linkage between multiple views across multiple analytic spaces. Furthermore, we support coordination and interpretation of multiple models.

The visual mining work in the literature concerning user experiences is also relevant to our work. Show Me [MHS07] proposed a query language VisQL that formalizes the transformation from data to visual representations. To automate the process, *Automatic Marks* are proposed to create rules for different data types so that views can be selected accordingly by algorithms. In MaVis, we automate the data reduction process and map the summarized information to the view space. No language is given, instead, we focus on a selected types of visual representations for data exploration. Visual aided diagnosis is another category of visual mining applications. Alsallakh et. al. [AHH+14] proposed several visualization techniques to visualize the multi-class classification confusion matrix so that the analyst may understand the source of errors. In MaVis, we instead focus on the diagnosis of local errors of a modeling process. For example, when a global trend is found over one year, the user may confirm whether the quarterly trends are consistent with it with ease.

## 2.5 Summary

In this chapter, we present the MaVis framework. It is a system designed for identifying co-movement patterns in time series dataset. It provides four analytic spaces that allow the analyst to navigate between them. It integrates multiple models to support the interpretation of the data space from multiple angles by comparing the different model types. MaVis also captures local dynamics of the time series data and allows the user to analyze connections between different time intervals. We evaluated our system with stock price data in a case study and also conducted user study measured the performance of subjects using our system.

# Chapter 3

# Model-driven Feature Analysis

A feature modeling and visual exploration system FeaVis is demonstrated in this chapter where the features are clustered based on feature similarities metrics. Techniques for visualizing the redundancies between similar features are discussed, additionally, analysis of features with partial similarities are supported. This work is mainly from an unpublished manuscript [ZWRH12].

Nowadays, it is common to deal with high dimensional data while performing data analytic tasks. On one hand, several automatic feature selection algorithms [ABK98, MMP02, YL03, YL04] are proposed for boosting up the performance of machine learning models by searching for the most suitable subset of features. On the other hand, a number of quality metrics [YWRH03, SS04, PWR04, JJ09, IMI+10b] have been proposed for ranking or reordering the features for maximizing interpretation of a dataset. Both directions yield promising outcomes for data analytic tasks but are limited to their own domains. This work instead investigates how to make use of the automatic feature searching strategy but now supported also by the visual analytic techniques to facilitate the feature space exploration.

The goal of FeaVis is to provide a visual analytic workflow for revealing the relation-

ships between features of the input data and identifying a subset of interesting features for further analysis. Both a clustering model and user-driven selections is provided to support interactive feature analytics. It enables the analyst to interact with a cluster model space of all the potential features. The appropriate relationship model of features are constructed based on the redundancies among the features established using pairwise metrics such as information entropy. The workflow is evaluated with real-world datasets collected and analyzed by financial experts, and this work concludes that the pairwise metrics as well as the clustering plus user-driven workflow is able to help the analysts to identify features of interest that are consistently used by published empirical studies which usually involve a long trial-and-error process.

## 3.1 FeaVis Workflow

The automatic feature searching algorithms and interactive feature selection methods share the same goal to find most appropriate feature subset for a high dimensional dataset. The found subset can be used either by a machine learning model (i.e., linear regression) or visual analytic views (i.e., parallel coordinates) to further support data exploration. This work aims to provide a generic framework that utilizes both automatic feature selection and interactive visualization support to offer flexible feature exploration. An overview of the system workflow is described in Fig 3.1 and explanation of each component is introduced briefly below.

- **Feature Clustering**
  - **Automatic Weighted Feature Clustering:** It is analogous to the data clustering algorithms such as DBSCAN [EpKSX96] and K-means [Boc07] in the sense that it groups features instead of data objects. However, the proposed feature clustering process is different from classic clustering algorithm in the

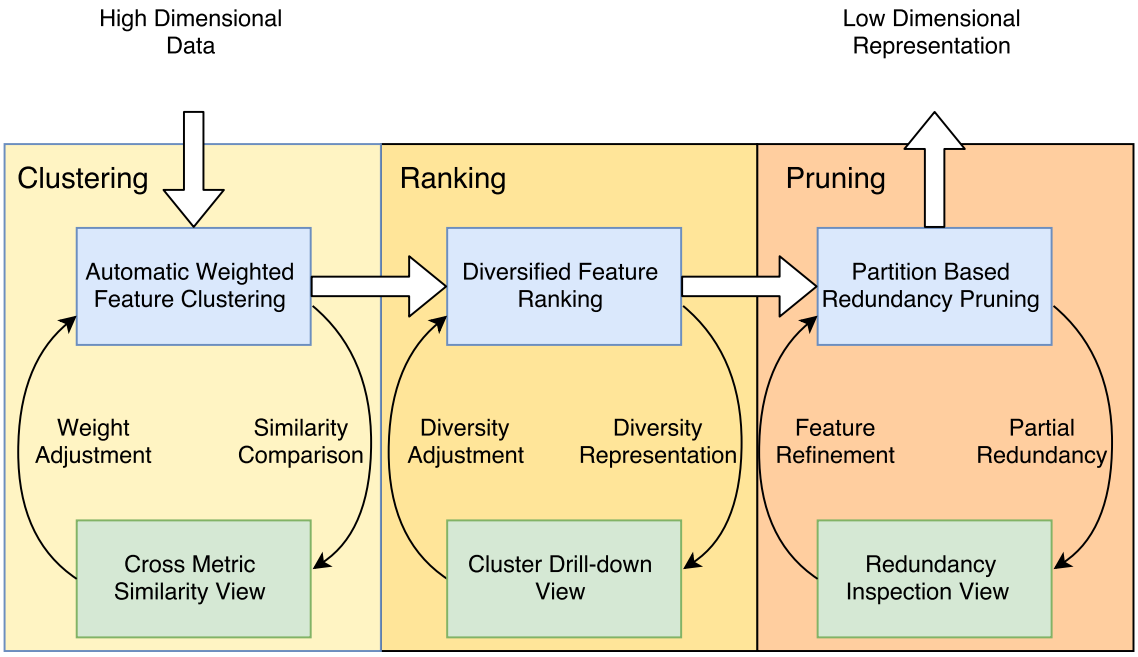


Figure 3.1: The overall workflow of the FeaVis system. The top 3 components are model-driven algorithmic methods that search the most descriptive subset of features based on given metrics automatically. The bottom 3 components are interactive visual support that help refine and interpret the automatic processes.

following: The distance metrics in this work are specific to feature similarities such as correlation, distribution similarity and cross entropy and thus is different from the commonly used clustering distance metrics such as euclidean distance or cosine similarity. Further, this work integrates multiple feature similarity metrics with a weighted aggregation to provide flexible feature relationship analysis given different tasks.

– **Cross Metric Similarity View:** To support multiple metrics for feature clustering analysis, this work integrates a line up view [GLG+13] to show the most similar features to a specified feature under different metrics. The line-up comparison is then used for spot checking whether the weighted clustering result makes sense to the analyst for her particular tasks. The corresponding

weight could then be adjusted to refresh the similarity metric for a updated line-up view.

- **Feature Ranking**
  - **Diversified Feature Ranking:** Many quality metrics (e.g., [SS04]) for measuring the importance of features ranking metrics exist. However, ranking metrics alone are not able to capture the interestingness of whole data space. For example, in the process of searching for a feature subset for a linear regression model, the top ranked features that are highly correlated to the target feature may likely have colinearity and thus lead to linear models with bad performance. This work focuses instead on a diversified feature ranking strategy that selects the top-k interesting features while maintaining a certain level of diversity so that the selected subset has less redundant information.
  - **Cluster Drill-down View:** To support the diversity and the ranking exploration all together the FeaVis workflow provides a Cluster Drill-down view to explore the feature relationship within each cluster. The features are placed in an increasing order in terms of their similarity to all other features of that cluster. Therefore an analyst is able to pick the ones are less similar to others (i.e., the more diverse ones) based on the view.
- **Feature Pruning**
  - **Partition Based Redundancy Pruning:** To further refine the redundant features in a cluster, the similarity between features in a cluster are further examined on different data partitions to detect partial redundancies. Each feature of interest is partitioned into smaller bins first and then for each bin the similarity between each pair of features are re-evaluated. This partition-based similarity

measure can further help to remove redundant features if similarity is present in a number of the partitions.

– **Redundancy Inspection View:** The view provides an overview of the partition-based redundancy. The redundancy inspection allows the analysts to determine if locally a feature is similar to another one even though the global relationship does not have high similarity. The view lays out every feature within a cluster and every partition on each feature is assigned a redundancy score based on partial similarity measure within the partitions.

## 3.2 Feature Clustering

With a similarity definition, the features can be grouped into different structures. The structure can be hierarchical [YWR02, YWRH03] or relative visual positions [YPH+04]. The similar features can then be refined to reduce the amount of information for displaying [MMP02, YL03, YL04, IMI+10b], or can be placed together to enhance certain visual presentations [PWR04, JJ09]. Grouping features is important in discovering interesting patterns in different ways. In this approach, a hierarchical clustering algorithm with Ward's minimum variance [WJ63] is used to cluster the data features. The benefit of this strategy is that the result from running the clustering algorithm better support interactive re-clustering which can be driven by the user while examining the feature relationship in the view this work provides.

Feature clustering is analogous to data clustering which classifies a collection of data objects into subgroups by computing the distance between data objects using metrics such as Euclidean distance. The feature clustering algorithm instead calculates the similarity measure between each pair of features. There are multiple ways to compute the similarities between a pair of features. Different similarity metrics define different types of

feature redundancies, such as linear dependencies, statistical properties, and information divergence. It is not possible to find a best similarity metric suitable for every possible analytic task. This work thus integrates 3 different similarity metrics, discussed next, to meet the needs of several analytic purposes. However, the framework is flexible to integrate additional or other metrics as needed. Currently the FeaVis workflow supports *Pearson Correlation* [Spe04], *Central Moment Comparison* [Ram02], and *Cross Entropy* [KL51]. Let us discuss the 3 feature similarity metrics first and then we introduce the weighted clustering algorithm.

### 3.2.1 Correlation Coefficient

Pearson Correlation is a well-known statistical method that is used in many visualization systems [JJ09, IMI+10b, PWR04]. The usage of correlation in these systems is to find correlated features and group them to serve different purposes. The correlation coefficient $\rho$ between $x$ and $y$ is defined as

$$\rho(x,y) = \frac{cov(x,y)}{\sqrt{var(x)var(y)}}$$

where $cov(x,y)$ is the covariance between features $x$ and $y$, and $var(x)$, $var(y)$ are variances of x and y respectively. The measure $1-|\rho|$ satisfies all the properties (positivity, reflexivity and symmetry) that a similarity metric must have [ABK98]. $1-|\rho|$ has the properties we need for clustering. The range of $|\rho|$ is $[0,1]$ where 0 and 1 indicate no correlations and strong correlations (positive/negative), respectively. We use $|\rho|$ here as both the strong negative correlation and strong positive correlation between two features suggest that they are redundant. In our system, we use a variation of correlation invented by Spearman [Spe04].

### 3.2.2 K-th Central Moment

Statistical properties of the features determine the quality of views formed by these features. In Rasmey's work [Ram02], moment-based strategies are used to determine the shape of one dimensional data that can be used to rank the interestingness of features. Similarly, Histogram Density Measure (HDM) [TAE+09] ranks the features based on how well data points are separated which is a more specific application of 1-D dimension ranking. We use more general statistics to measure each feature, as here we are more interested in the general shape of each feature such as skewness.

The K-th central moment is a mathematical measure of a given statistical distribution, represented as:

$$m_k = \frac{1}{n}\sum_{1}^{n}(x_i - \mu)^k$$

where k the the degree of moment, n is the sample size and $\mu$ is the mean. Since the first central moment is 0 and does not contain any useful information, we use the mean value instead. The second central moment is variance. The third central moment measures skewness, which represents the symmetricity of a distribution. The fourth central moment measures kurtosis which represents the shape of the shoulder and tails of a distribution. We use up to the fourth moment in our system, as higher order moments characterize the shape of the distribution in more abstract ways and are not visually perceivable. We scale each feature to the range of $[0,1]$ then we generate a feature vector of size 4 composed of the mean value, variance, skewness and kurtosis of the feature. Then we calculate the distance matrix between each pair of such feature vectors using Euclidean distance (other distance metrics for the feature vectors can be plug in in the future). Thus K-th Central Moment similarity metric effectively measures how the shape of the distribution of each feature is different from others. A smaller distance between two feature vectors means the two corresponding features have similar statistical shapes. Using this similarity metric,

we are able to group the features with similar distribution shapes (e.g. bimodels) together that may indicate some degree of similarities in terms of the real world processes that generate the data.

### 3.2.3 Cross Entropy

Cross Entropy (or Mutual Entropy) is a measure of the mutual dependency of two random variables in the information theory literature [KL51]. The definition of cross entropy is based on the definition of Shannon Entropy ($H$):

$$H(x) = E(-ln(x))$$

where $E$ is the expectation and $x$ is the input feature. The cross entropy between dimension $x$ and $y$ can be represented as:

$$H_c(x,y) = \frac{H(x) - H(x|y)}{H(x) + H(y)}$$

$H_c(x,y)$ is proven to be symmetric and is bounded to $[0,1]$ in [YL03]. The cross entropy is also known as KL divergence [KL51]. It is used in a visual analytic application [SSN+11], where the metric is used to measure the distance between two distributions. In our approach we also binned the continuous variables before calculating KL divergences. A cross entropy of 0 means that given one feature, no extra information is needed to describe the other dimension. When two features are distinct, $H_c$ approaches 1. In machine learning domain, this property is often used as a metric to identify how well one feature predicts another one that is considered one other type of similarity in this work.

### 3.2.4 Automatic Weighted Feature Clustering

A single metric is often only useful for a limited scope of exploration. For example, the correlation metric is able to detect linear correlations between a collection of features. However, in many situations, analysts may want to explore the feature space to identify the most descriptive features in the high dimensional space. Combining the metrics [JJ09] can be a promising way of supporting interactive user metrics. Our work not only supports interactive user metrics, but also enables metric comparison and refinement for feature clustering.

The most important question this work tries to answer is, *how do different feature similarity metrics impact on the feature clustering process?* In order to interpret the clustering result under different metrics, FeaVis workflow uses a weighted clustering strategy and allows the analyst to evaluate the effectiveness of each similarity metric interactively. The main challenge of interactively comparing clustering results is to compare multiple clustering results all together in a visual representation to support interaction based on per-computed similarity relationship. Typically, a clustering process needs at least one scan of the dataset which requires $O(N)$ runtime where $N$ is the number of data points. [XW$^+$05]. The time for computing the distance between data points is often insignificant as the number of features $d$ is often much less than the number of data points $N$. However, for the clustering process in FeaVis, the time for computing distances between different features is not insignificant. It is determined by the size (number of data points) of each feature which is often large.

To support the real-time querying and analyzing the feature similarities across different metric spaces, FeaVis uses a automatic weighted clustering strategy that aggregates a set of feature similarity metrics. It allow the user to query similar features to one feature of interest with the flexibility to specify what similarity metrics are more important. In the meantime, it also help analysts to determine what user metric can discover interesting

feature relationship by examining the feature similarity generated by a certain user metric that is combining several predefined metrics.

Then we discuss a pre-computing and optimization strategy for interactive metric tunning. It supports weight tuning and provide a local verification mechanism for visual feedback without re-cluster the features. A naive way of finding clusters based on a new user metric is to combine the similarity metrics first to compute similarities such that similarity $S_{agg}(X,Y) = \theta_1 \cdot S_1(X,Y) + \theta_2 \cdot S_2(X,Y)$. Then the clustering process is executed according to the computed pairwise relationship by $S_{agg}(X,Y)$. If a analyst wants to adjust the weight during the analysis phase, the clusters have to be re-computed for the user metric with new weights. Such pipeline is inefficient and FeaVis instead caches the similarity by predefined metrics such as $S_1(X,Y)$ and $S_2(X,Y)$ and then perform similarity search. The similarity search and visualization is performed by focusing on a user selected feature and its similarity to other features. The similar features of that given feature can be examined by only finding and visualizing the $d-1$ pairs of relationship instead of $d \cdot (d-1)$.

**Similarity Matrix Pre-Computation**

In this subsection we explain in more detail how we handle the feature similarity computation. In FeaVis, the user metric between feature $X$ and $Y$ is calculated based on a weighted sum:

$$sim_u(X,Y) = \sum_{i=1}^{3} \theta_i sim_i(X,Y)$$

where $sim_u$ is the aggregated user metric, $sim_i$ is an individual metric such as correlation and $X$ and $Y$ are two features.

The concept of caching is to store the similarity of each pair of features regarding $X$ and $Y$ and a tuple of user weights $\langle \theta_1, \theta_2, \theta_3 \rangle$ so that similarity based on any user metric

regarding a set of weights can be acquired by using cache function $f(X, Y, \langle \theta_1, \theta_2, \theta_3 \rangle)$ in constant time after adjusting the weight parameter $\theta$. The computation of user metric in this case can be very efficient. In reality the weight space can be huge as the weight can be in any arbitrary granularity (e.g., 1E-6 at the scale of $(0, 1)$) which may use a huge amount of caching space. To tackle this problem, the weight are at a specified granularity 0.1 as default so that the total number of combinations becomes an *n choose k* problem where $n$ is 10 and $k$ is 2. It is equivalent to the problem of distributing 10 identical balls (total sum of weight divided by default granularity) to 3 buckets (three predefined metrics). Adding more metrics may require a finer granularity that is a caching problem that is out of the scope of this work.

**Similarity Verification with Sorted Neighbors**

The next problem to be solved is to provide a feedback loop so that the goodness of the customized user metric can be evaluated effectively by analysts through the channel of visualization.

FeaVis supports spot checking the user metric and provides feedback to the user metric weight setting by providing a comparison system that allows comparing feature similarities among multiple metrics. Specifically, the spot checking compares the three sets of $k$ most similar features of any given feature $X$ for the three corresponding metrics.

For example, for any user specified feature such as *total assets*, three lists are generated based on the cached similarity matrices of the 3 predefined metrics. If analysts have any knowledge about this feature, they may contribute their knowledge to the process by examining the three sorted list of most similar features and then adjust weights based on their preferred lists. Then the generated new user-metric may be used to another iteration of generating similar features based on the weighted aggregation. Since the iterative adjustment is based on the cached pairwise relationship, it allows analysts to efficiently

explore the metric space as well as the feature space.

Next, we discuss how the features are clustered based on the predefined and customized metrics. As observed in Table 3.1, the features can be similar to a specified feature (i.e., total assets) in a set of metric spaces. The three similar features based on the correlation metric are *total debt*, *total sale* and *total profit*. Combining the three metrics reveals that *total sale* and *total profit* are more similar to the selected feature: *total assets*. However,the plain table view is not able to effectively guide the user to digest and contribute into the customization process of user metric.

| name | correlation | distribution | cross entropy | user metric |
|---|---|---|---|---|
| total debt | 0.83 | 0.42 | 0.51 | 0.59 |
| total sale | 0.71 | 0.59 | 0.73 | 0.68 |
| total profit | 0.65 | 0.68 | 0.51 | 0.61 |
| ... | ... | ... | ... | ... |

Table 3.1: Example of similar features for feature *total assets*. By default the aggregation weight is 0.333 for each metric and the similarity is normalized to (0,1).

### 3.2.5 Cross Metric Similarity View

Next, a visualization strategy is introduced to support ranking of similar features by multiple attributes (i.e., the multiple metrics for measuring similarities). The cross metric similarity view is provided to compare and contrast the ranking result from multiple metrics and allow the user to determine which features are more interesting based on a selection of metrics. With the predefined metrics that are discussed earlier, an analyst is able to hand craft user metrics by adjusting the weight of combined metrics and then observe an refined similar feature sets in the *cross metric similarity view* (Fig 3.2). The main purpose of this view is to guide analysts in creating a combination of weights for a user metric that is appropriate for their tasks. Then how to adjust the weight for the predefined metrics is the main problem to be solved here.

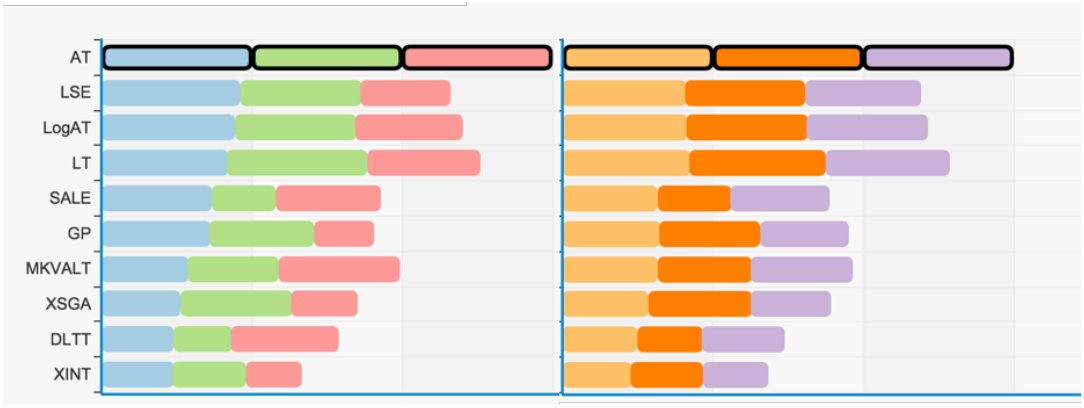


Figure 3.2: The view shows a comparison between the 3 default ranking metrics (left) and 3 user metrics generated by combining the 3 default metrics (right). The user metrics are generated by using different weight combinations, in this case [0.5,0.3,0.2], [0.1,0.8,0.1] and [0.4,0.2,0.4] respectively. The feature on top is the focused feature and it has similarity score of 1 to itself. Other features are ranked based on similarities to the focus feature using different metrics. The length of bars represent the similarity score. (AT: total assets; LSE: leverage; LogAT: log total asset; LT: total liability; SALE: total sale; GP: gross profit; MKVALT: market value; XSGA: general expenses; DLTT: long term debt; XINT: interest expenses.)

To guide the weight adjustment, the *corss metric similarity view* provides: 1) Ranking comparisons between different predefined metrics; 2) The user metrics that are generated by combining the predefined metrics using different weights. Inspired by the ranking strategy in [GLG+13], this work incorporates a multiple ranked lists in the view.

For example, when an analyst is investigating a dataset before building linear models, she may want to emphasize on the correlation metric by assigning a larger weight to it and use smaller weights to the other two metrics. The question is what is an appropriate weight setting? In FeaVis, any user specified combination of weights is evaluated in the *cross metric similarity view* by providing a similarity ranking list of features (Fig 3.2). In this view, FeaVis calculate the resulting ranking list against existing user metric settings to detect if the new weights generate a new ranking list. The the weight settings that lead to new results are preserved in the view space.

The *cross metric similarity view* implemented in this system allows analysts to iter-

atively adjust weight for each predefined metric and this way generate a weighting that results in a new user metris for comparison. Such generated user metrics are used to cluster the features for further analysis.

## 3.3 Feature Ranking

Previously, the features are clustered into similar groups to help the feature similarity explorations. However, in many situations, the features within each group may have different importance scores which implies the selection of features within a group is not a trivial task. The degree of importance of one feature can be measured by, for example, how representative it is in a group of features. Many different ways of determining the degree of importance can be found in [BTK11]. The focus of this work is to rank the features while considering diversity among them. During the ranking process, the feature importance is calculated based on how close they are to each other and ranked in a spectral space (Fig 3.3) as explained below.

### 3.3.1 Diversified Feature Ranking

By default, the most important features within the group are selected into the descriptive data subspace. The main goal of this component is to rank and diversity the features to help select most representative features. To support feature ranking, we use several measures to help diversify the selection, namely, *center-based metric* and *amount of outliers*. In the mean time, to support the diversifying process, we design and implement a *feature neighborhood view* (Fig 3.3) to show redundancies within a cluster of features. The manual selection is supported by FeaVis to override the default ranking metric if the analyst feels the features that are suitable for her task is not selected by default. Then she can verify if her hypothesis stands by examining the neighborhood of the selected feature

and the overall distribution of features within that cluster. We discuss our methods in the following paragraphs.

### Ranking Features by Importance

First, the ranking metrics that are integrated in this work are discussed. In a *center-based metric* the analyst is able to rank the features in the order of how close one feature is from the center of the cluster. We define center as

$$\underset{X \in G}{argmin}(\sum f(X,Y)) \quad \forall Y \in G$$

where $f$ is one of the similarity metrics we mentioned in Section 3.2 and $G$ is the group of features. This metric considers the features more interesting when they appear closer to the center of a feature cluster. The semantics behind this metric is that the features closer to the center are more similar to the rest of the group. Obviously, the features at the boundary of the group is less similar to all the other members of the group. [BvLBS11] uses a similar idea to filter the features in a redundancy group.

*Amount of outliers* is another interestingness measure for the features. Following [WAG05], we choose the $\omega_l$ and $\omega_u$ as the lower and upper thresholds for determining outliers on one feature. $\omega_l$ and $\omega_u$ are defined as

$$\omega_l = Q_1 - 1.5 * (Q_3 - Q_1)$$

$$\omega_u = Q_3 + 1.5 * (Q_3 - Q_1)$$

where $Q_1$ and $Q_3$ are lower and upper quartiles of this feature. We use the proportion of data instances outside the range of $[\omega_l, \omega_u]$ as the outlying score.

### Diversifying Selections

In this view, each column represents one feature. The k-th row from the bottom in each column represent the k-th nearest feature to that feature. Each color grid is used to present the similarities between the current feature and its k-th nearest neighbor. The color grid is arranged so that the farthest neighbor appears at the top of that column and the other grids follow a descending order. Thus, the first neighbor appears at the bottom of the column. The ordering of the columns are based on the result of a ranking metric.

This work uses a *feature neighborhood view* (Fig 3.3) to illustrate the redundancies among the cluster of features. In this view, the highlighted column highlighted indicates the default representative of this group that is prioritized by the default ranking metric. The spectrum of that column indicates how close that column is related to its neighbors shown as colors vertically where each color cell represents a similarity score to its neighbor. Based on this view, we can see the columns to the left are close to other features except two also the columns to the right are more distinct from other features. To avoid selecting redundant features and manually override the default selection to represent the whole group, a good strategy is to select from the right side of the plot.

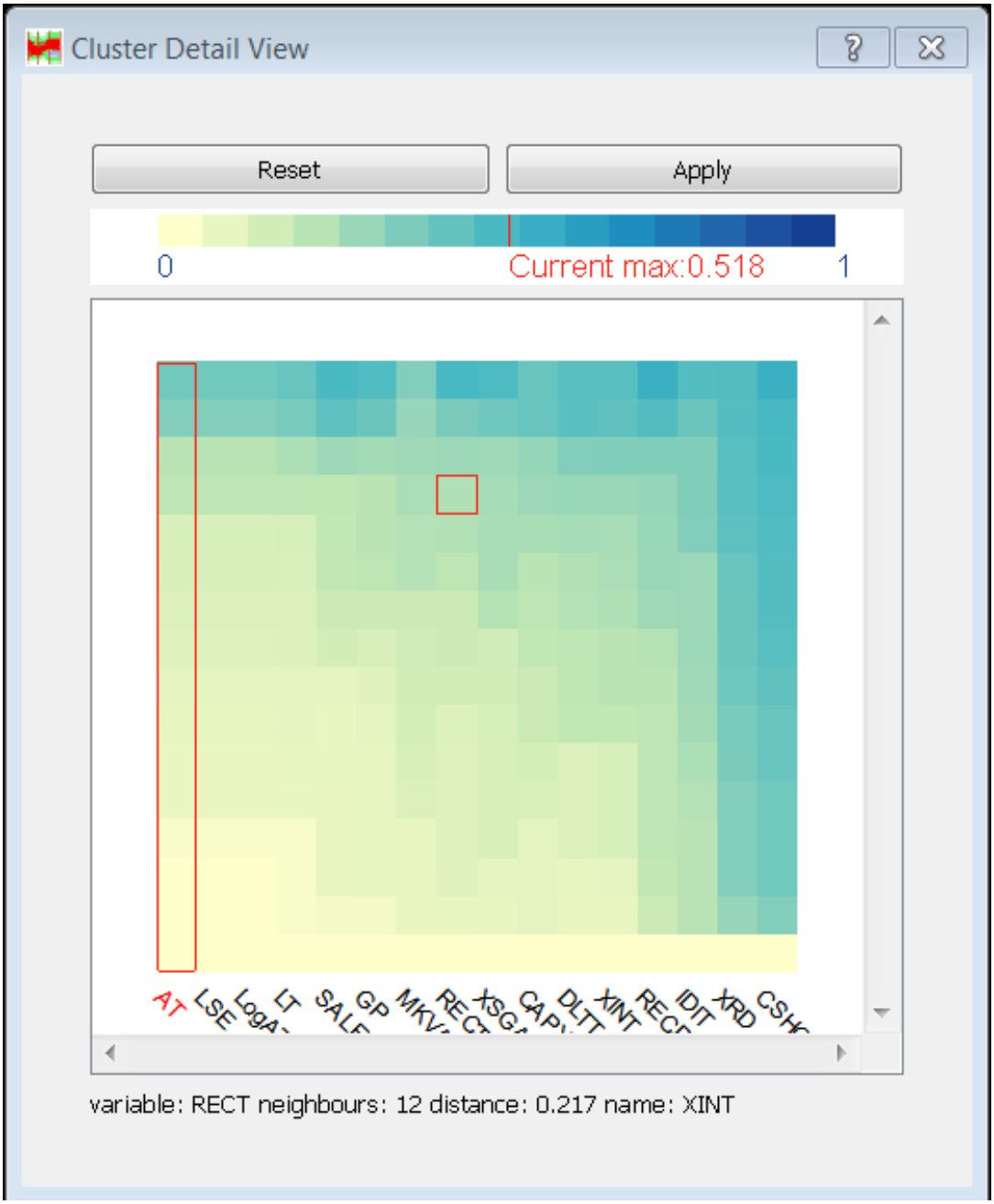


Figure 3.3: The detailed view of a cluster of features. The column represents a feature, and for each column the color of a grid indicates how far this feature is away from its neighbors. The first column is automatically selected as a representative of this group (long rectangle). The small red selection box to the right in the view is a cursor over selection which shows more information about that particular neighbor.

To further help examine the diversity, we also shows more information about the

neighbors of one specified feature by mousing over the corresponding cell (shown as the smaller rectangle in the view).

### 3.3.2 Feature Cluster Drill-down View

Following the workflow and three main computational components described earlier, we now discuss the design decisions made for the visual representations that guide analysts for exploration.

The design of the cluster view is inspired by the VHDR system [YWRH03] and the Interring system [YWR02]. The feature hierarchy in these two systems can help the analyst to interactively select/brush features of interest to do further analysis in a lower dimensional space. The hierarchical representation uses derived features to represent the underlying similar features. In some domains, such as financial analytics, analysts prefer features that are collected or computed by other experts that actually carry important meaning. The derived features commonly used for visual representations are often hard for the analysts to interpret. Instead, in our approach, we thus use a cluster view that is based on the hierarchical structures generated by the algorithm and similarity metrics we discussed in Section 3.2, Section 3.3 and Section 3.4. Although the hierarchical structure is not displayed in our view (Fig 3.4), we allow the analyst to control the clusters by supplying a cutoff value. In the meantime, the statistics of each group are updated and displayed. We also considered incorporating the summary statistics of each group into the hierarchical view, but decided the resulting view was cluttered and less scalable. After such considerations, we chose to use a scatterplot and profile glyph (Fig 3.4). The layout of the scatterplot is computed using an MDS algorithm [CC00] offered in R [R C12]. The profile glyphs are used to show summary statistics of each cluster.

The statistics we provide to the analyst include 1) the size of the group, 2) the average distance, 3) the variance of distance and 4) median distance between any pair of mem-

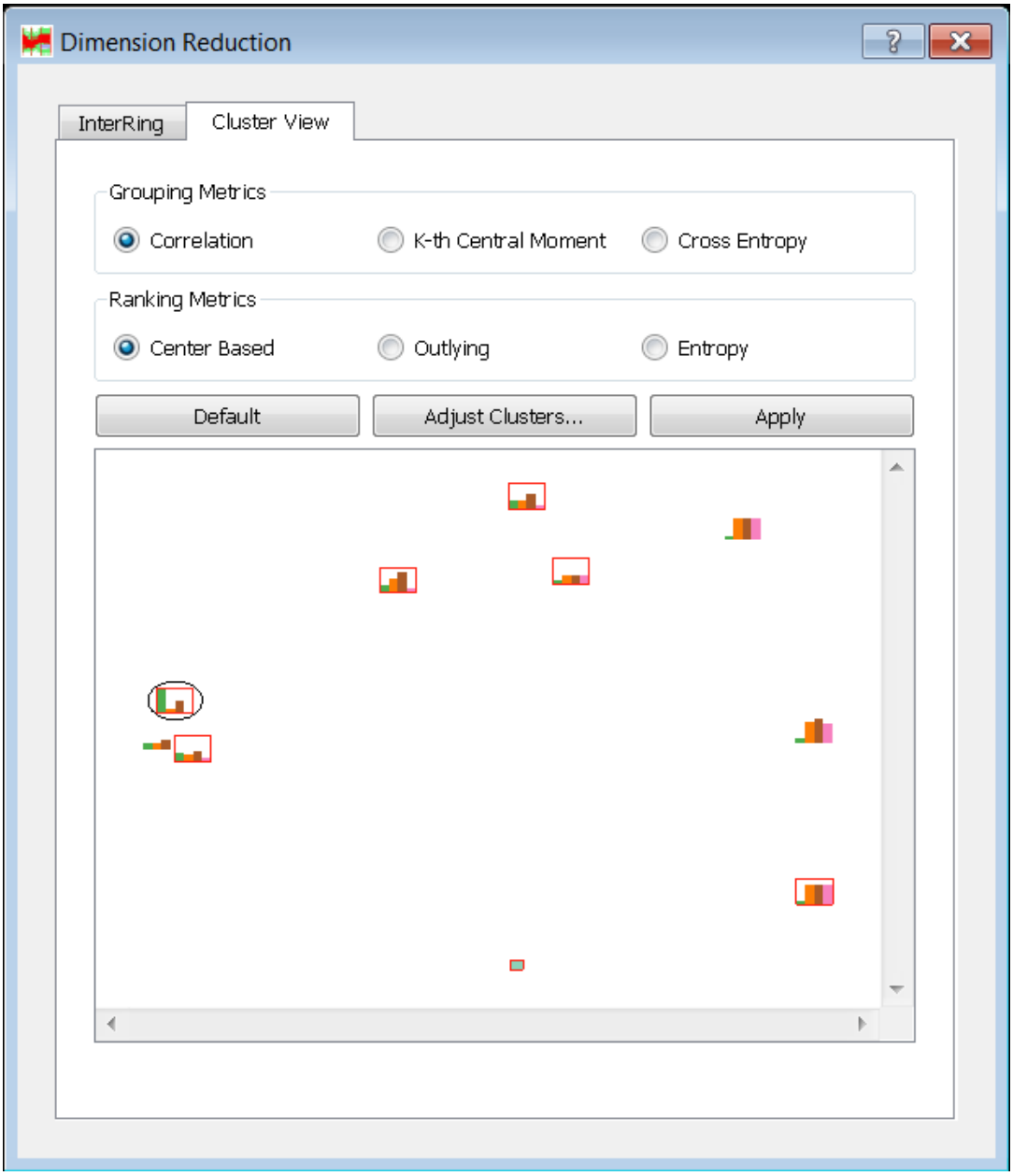


Figure 3.4: Cluster view of 45 features in 10 groups, including one single element group represented by a cyan rectangle. The group can be selected/unselected and the selections are marked with small red boxes. The black circle over the group indicates a marked focus group by an analyst, the details of the focus group are displayed in a different view, shown in Figure 3.3.

bers. The four measures provide an overview of each feature cluster that help analysts to identify cluster of interest for drilling down analysis. The interestingness of glyph shapes

are illustrated in Fig 3.5.

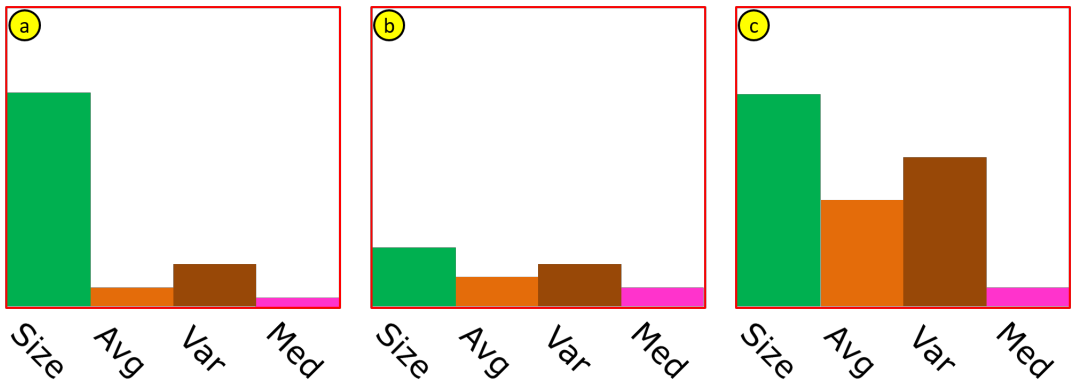


Figure 3.5: (a) A relatively large feature group with high in-group similarity, indicated by the relatively low average distance, as well as low variances. It indicates the large group of features are very similar to each other. Thus the redundancies in this group is significant. (b) Based on the same reason, b shows high intra cluster similarity but it is a much smaller cluster. (c) It is a relatively large group with low in-group similarity. The confidence of removing redundancies in this group using automatic methods is less for the group on the right.

The cluster on the right is worth further investigating as the degree of redundancy is relatively higher than other two clusters. The other two examples show a large and small cluster with a fair amount of redundancy. The hierarchical clustering cutoff parameter is controlled by the analyst which is used to generate clusters with specified distance range.

There are at least two good ways of implementing the glyph layout [War02]: (1) Place the glyphs based on the glyph similarities so that feature clusters with similar intra cluster similarities are placed at neighboring locations; (2) Place the glyphs based on inter feature similarities, so that the groups that have similar features are near each other. The advantage of (1) is that the feature clusters of interest to a particular analyst are neighbors in the view; it can speed up the exploration as the analyst is able to identify similar glyphs in a relatively small region. The layout (2) can tell the analyst how similar any two groups are based on their relative positions on the screen, while the layout (1) does not provide such information. Hence, during the interaction of adjusting the number of groups the

analyst is aware of what an appropriate stopping point is based on the distribution of the glyphs in layout (2). We considered implementing both but switching between two layouts loses context. This hinders the ease of the exploration process. We thus proceeded with design (2) after considering the advantages and disadvantages. The key reason is that the patterns found in (1) can be seen in (2) given some extra time (by recognizing the shape of the glyphs) but some patterns (i.e., inter cluster relationship) in (2) can not be seen in (1).

Figure 3.4 shows 10 groups and one single element group (represented as a small rectangle). The red box outside of each glyph indicates the group is selected as descriptive. The analyst can select and unselect any glyph by a single click. Unselecting one glyph means all the features of that group are considered not interesting and thus are removed from the descriptive subspace.

## 3.4 Feature Pruning

In this section, this work primarily focuses on identifying redundant features from the stability perspective as some of the redundancies may only exist in certain subsets of data space. Partitioning the data space on the features can help the analyst identify local correlations. Such local patterns indicate whether the local redundancies exist that may not be captured by the clustering and ranking methods discussed earlier.

### 3.4.1 Partition Based Redundancy Pruning

Next, a partition based redundancy pruning and inspection method is introduced to support feature pruning in a local subset data space. The descriptive features we get by applying metrics globally on all data instances may be less meaningful for some subsets of the dataset. In order to investigate the local redundancies to determine a good set of

features on different subsets of data points, we first partition the data into smaller subsets and apply the similarity metrics to each partition to get local similarity. Then the stability of the similarity relationship between different features is computed across all the partitions.

The approach in [MBD+11] offers two partitioning methods, which we think are typical in subsetting high dimensional data: 1) Partition all features based on one feature; 2) Partition each single feature based on the values of that feature. Neither of these satisfies our need. The local patterns we are interested in are redundancies between features. Thus correlations between features should be presented to the analyst after the partitioning. Method 2) is not able to show the relationships between the features locally, because the partitioning on each feature is independent of each other. As for method 1), it can be effective if the one feature we choose to partition on has good local structures, but the analyst may have to exhaustively search the possibilities. Another way of partitioning high dimensional data on all features is to iteratively embed the features as in the dimensional stacking display [LWW90]. The downside of this method is that the size of bins after combining all features is small and hard to control. Also, the number of bins could grow to $b^d$, where $b$ is the number of bins on each feature and $d$ is the number of features. After considering the above alternative options, we decided to use a parallel partitioning method where the data space is partitioned on all features one by one, and the partitioning result is saved for each feature. The partition strategy of this process is shown in Figure 3.6 where the top path shows partitioning on Dim 1 and the bottom path shows partitioning on Dim 2.

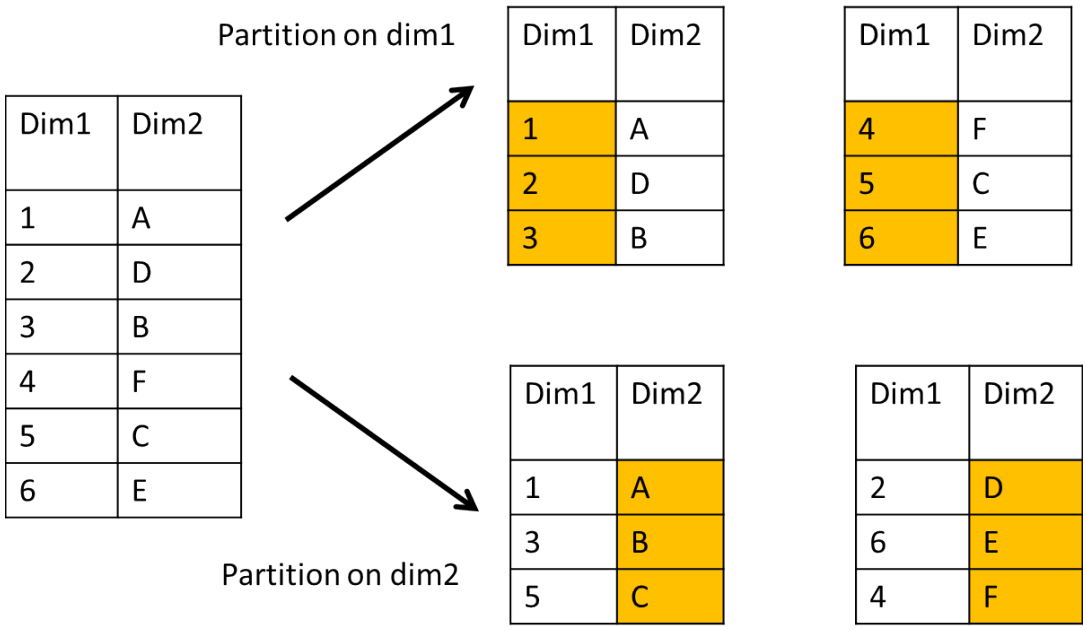


Figure 3.6: Illustration of process for partitioning on two features. The bin size is 3 and the number of bins is 2.

For each feature we partition on, we use equal count binning, where each bin has the same amount of data points. Using equal width binning can certainly be an alternative option, but enumerating all possible binning methods is not our primary goal. The main idea of our work is to show the interestingness of the local patterns, not how best to define the semantics of ”local”. The local patterns we show in Fig 3.7 are based on equal count binning.

### 3.4.2 Redundancy Inspection View

The inspection view is mainly used to inspect the stability of the feature similarity relationships over different data partitions. To present the stability of the pairwise similarity is a challenging problem in that the number of possible partitions is large. For a cluster of features of size $k$, the pairwise relationship is $k^2$, the complexity increases to $k * p * k^2$ where $p$ is the number of partitions on each feature. To quickly guide the analyst to

the most relevant information, the *redundancy inspection view* aggregates the similarity relationship before visualizing the stability.

The similarity between each feature and all other features are calculated and summarized using average similarity. The similarity between feature $X$ and all other features in $\{S-X\}$ are averaged to a summary description to measure the similarity of $X$ to other features in the cluster $S$. Then, each partition of $X$ is measured against the corresponding partition of all other features to generate a list of local similarity averages.

The next step is to visualize and compare these local similarities with the global similarity measure. In order to gain confidence before pruning any features from the final selection, an analyst still needs to examine the stability of the relationship between one feature and the others. FeaVis compare the local similarity to the global similarity by plotting the local similarity against the global similarity.

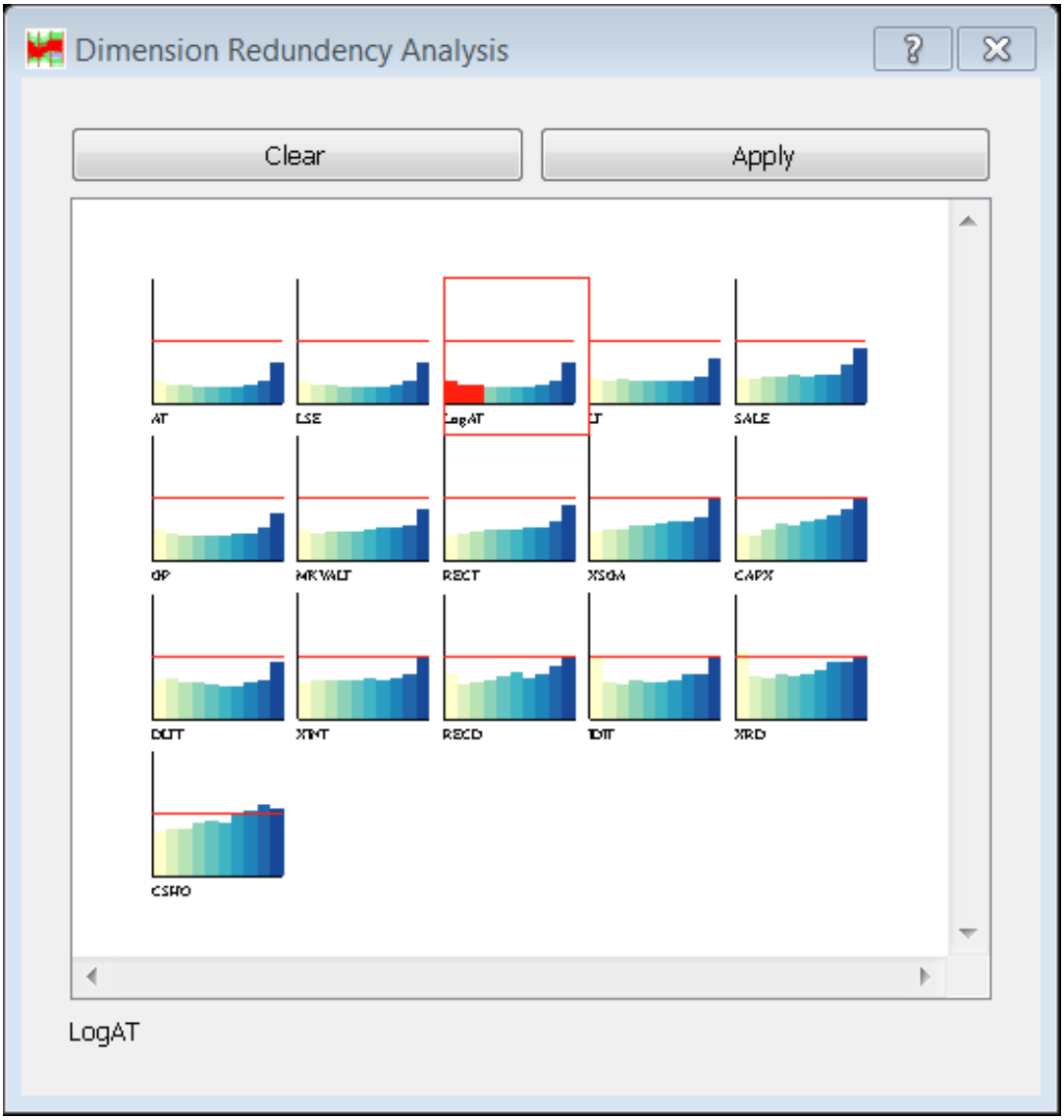


Figure 3.7: The analysts may examine the stability of the feature similarity across every partition. In this view, each histogram view represents the stability of one feature vs the others within a cluster. The horizontal red line indicates the global similarity between the given feature and the others. The label underneath each histogram represents the name of the given feature. The x-axis of each histogram represents partitions generated on the given feature arranged from low value to high value from left to right. The y-axis represents the degree of redundancy from low to high. The shape of the histogram represents the stability based on how close the bars are to the red base line.

To identify features to be pruned based on the above information, an interactive exploration method is required. In Fig 3.7, an analyst is able to determine the relatively more

redundant features are the features on the bottom 2 rows where the stability across different partitions is high and they are close to the global similarity measure. Discarding these features lose less information as the similarity relationship of those features are stable in the local partitions relative to the global space. The top two rows are not ideal candidate to discard. The redundancy score of local partitions are lower which means the features are less similar in local partitions.

## 3.5 Evaluation

In this section we introduce how to use our system through examples. The examples we use are in the financial domain. However, our system is not restricted to this particular domain and other datasets have also been tested. The purpose of using a domain specific dataset is to get feedback from experts and compare our visual representations with published empirical studies. In this domain, analysts often deal with modeling problems, such as identifying the characteristics of high risk. One of the biggest challenges in the modeling process is to find an appropriate subset of features to use. Typically, their selection process is based on the domain knowledge of an analyst. It relies on experiments or results from other studies. Few published works discuss methodologies that lead to a systematic way of choosing features. This section shows our system is able to facilitate the selection process by suggesting features that are similar to the ones identified in empirical studies. We also show that alternative features can be selected in some subsets of the data space to improve results. Through this case analysis, we show that our real-time interaction framework can achieve similar outcomes to the studies that require significant efforts by domain experts. The use case studies we use focus on building models to detect abnormal financial activities.

| Literature | Database | Explanation |
|---|---|---|
| DEBT/EQ | DEBTEQ | Debt/equity |
| SAL/TA | SALTA | Sales/total assets |
| NP/SAL | NPROFSA | Net profit/sales |
| NP/TA | NPROFTA | Net profit/total assets |
| RLC/SAL | RECSA | Receivable/sales |
| WC/TA | WCAPTA | Working capital/total assets |
| GP/TA | GPROFTA | Gross profit/total assets |
| INV/SAL | INVSA | Inventories/sales |
| TD/TA | DEBTTA | Debt/total assets |
| LAT | LogAT | Logarithm of total assets |
| | ZSCORE | Altman zscore |
| | EBIT | Earning before interest and tax |
| | ARchange | Account Receivable change |

Table 3.2: Explanations of features

### 3.5.1 Data Description

We extracted 45 features from Compustat [SP12], a database of companies in North America. The 45 features include those used in 3 financial studies, along with some miscellaneous features suggested by domain experts. Each feature is a measure of the financial status of the companies; for instance, such measures can be the sales, gross profit, and working capital. The features selected by financial experts are usually used to distinguish abnormal financial activities (e.g. falsifying financial statements) from normal ones. In Spathis's study [SDZ02], there are 10 chosen features. The notations in the database and our system are different from that used in Spathis study. We have a translation in Table 3.2, which also includes features in other studies [Suy09, Alt12, KSM07]. The number of companies we extracted from this data base is $3,791$.

### 3.5.2 Case Study: Representative Financial Variables

In this section, we show a walkthough of our system. The analyst loads the dataset we described into the system.

By default, every feature is visible to the analyst and visual patterns are hard to perceive. We start a reduction pipeline. The 3 views we described in Section 3.2 are shown. The cluster view shows several groups of features with summary statistics for each group (Figure 3.4). The cluster detail view (Figure 3.3) and the local view (Figure 3.7) show the largest group by default. In the meantime, selected features by analysts are visualized in a parallel coordinates view (Figure 3.8).

The next step is to refine this view by adjusting the number of groups and changing the representatives of any group if the analyst feels better representatives exist. The analyst starts to adjust the size of clusters by changing the cut-off value of the hierarchical clustering result. The more clusters she allows, the less redundancy to be removed. The extreme case is that every features is a single element cluster and thus by default all of them are selected as representatives of themselves. In this case no redundancy is removed. The other extreme case is only one feature is left, representing all other features. The more meaningful cases are when the analyst forms several groups and the number of groups is close to the number of features she wants to handle.

After the adjusting, she can continue with fine tuning or go ahead with the automatically selected features within each group. One group representative is shown at the bottom of the screen when the cursor is over any profile glyph in the cluster view. Some group representatives (i.e. AT) may not make much sense to her. In that case, she can double click on one of these groups and go to the detailed view of the group. Every member of that group is displayed in the cluster detail view (Figure 3.3). The analyst may feel that "LogAT" better represents the group; she then disables the red box over features "AT" by clicking on the column "AT" and enables the feature "LogAT" by clicking on the column "LogAT". She can modify other groups with manual tweaking and then stop to investigate the data space view (Figure 3.8). She can also further fine tune the model by looking at the local properties of a certain group as in Figure 3.7.

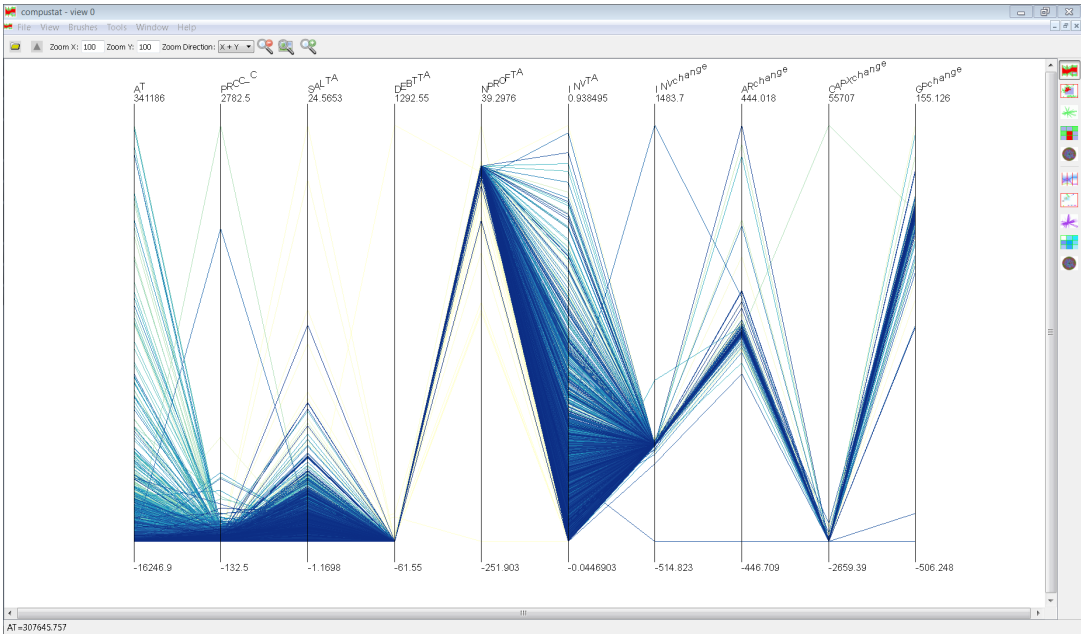


Figure 3.8: This view shows the features selected by global redundancy measure when the analyst finishes adjusting the number of groups. This selection is done without conducting any local redundancy analysis.

In the local view, the group features are less similar to each other in certain partitions (e.g., data points with low values in "LogAT") than in the global space. This may indicate the group members may be less similar in these partitions. The redundancy she removes based on the global similarity may mislead her to sub-optimal selection of features for these partitions. She brushes over the histogram "LogAT" and marks these partitions (Figure 3.7). A different grouping of the features is formed based on the similarities between features in the portion of the data she selects. She can switch to the data space view (Figure 3.9) to check the features that are representatives of the groups for the subset of data she selects.

### 3.5.3 Comparison to Empirical Studies

We compared the features picked by analysts over the workflow of the FeaVis System to the three published empirical studies [SDZ02, KSM07, Suy09]. The automatic process

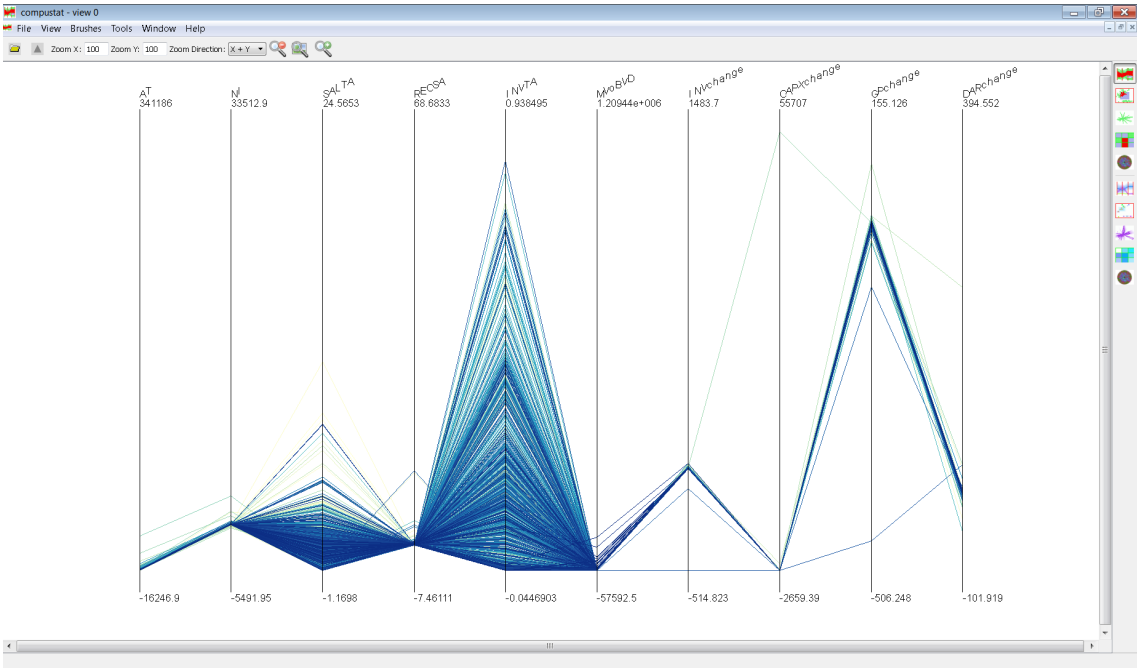


Figure 3.9: This view shows the features selected by local redundancy measures over a subset of data points. This view is generated after the analyst brushes the partitions of interest.

identifies several feature groups (represented as several sections in Table 3.3) and selects one representative feature out of each group by default. The selected features cover all the feature groups as we can see in the table. Using our dataset that is collected by analysts, some of the empirically chosen features are redundant as they appear in the same feature group. It is very common problem the empirical studies often encounter. When an analyst deals with a different dataset, the guidelines provided by the literature may not be perfectly appropriate. We show the level of redundancy with $1 - |\rho|$ ($\rho$ is a correlation score) in Table 3.3. Moreover, our system help identify the 3 core features that are used in all the three studies. It indeed offers the analyst a relative good starting point. She can choose to interact with the system and adjust the selection towards an improved descriptive subset of features. Two of the empirical studies use redundant features for their modeling process. Likely, the reason may be that domain experts want to emphasize the contribution of a particular feature group in a particular task. The other reason could

be that the dataset collected for this work has different redundancy relationship.

Another interesting observation is that for companies with lower values of "LogAT" (smaller firms) the most informative features shown in Figure 3.9 are not the same as the features we get globally in Figure 3.8. Treating firms of different sizes differently is a common strategy in financial analytics. Our local view provides them a tool to investigate any partitions on any feature.

| Name | [SDZ02] | [KSM07] | [Suy09] | Our pick | |
|---|---|---|---|---|---|
| SALTA | x | x | x | x | |
| GPROFTA | x | x | | | 0.38 |
| NPROFTA | x | x | x | x | |
| NPROFSA | x | | | | 0.12 |
| EBIT | | x | | | 0.15 |
| DEBTTA | x | x | x | x | |
| DEBTEQ | x | x | | | 0.39 |
| ZSCORE | | x | | | 0.18 |
| INVTA | | | x | x | |
| INVSA | x | | | | 0.16 |
| WCAPTA | x | x | | | 0.59 |
| WCAP | | x | | | 0.70 |
| ARchange | | | | x | |
| RECSA | x | | | | 0.78 |
| AT | | | | x | |
| LAT | x | | | | 0 |
| COSAL | | x | | NA | |

Table 3.3: The mark "x" indicates selection of that feature. The numbers in the last column are the measures of correlation $(1-|\rho|)$ between the selected feature and unselected features in a group. NA means there is no such feature available in our dataset.

## 3.6 Related Work

Strategies for finding lower dimensional projections of interest have been discussed in many works [AWD12, AEL+10, BM01]. In these works, the primary task is to project the original high dimensional data into a lower dimensional space that is human interpretable.

These approaches are similar to our approach in the sense of searching for lower dimensional representations of the original high dimensional data space.

Another related approach [IMI+10b] combines similar features together and use synthetic features to represent the joined features. The above techniques generate a lower dimensional data space through matrix transformations and other computations. The features generated by such computations do not have any readily understood meaning. Therefore, it is more challenging for the domain experts to interpret the result. In our approach, we focus on keeping the data semantics in the lower dimensional representations.

In searching for meaningful structure in a high dimensional dataset, quality metrics [JJ09, BTK11, PBH08, SS04, TAE+09, PWR04, WAG05] are used to measure the interestingness of the features. Dimensions (1D) or combinations of dimensions (2D and higher) are promoted or demoted based on the interestingness score assigned by the metrics. These approaches are able to identify subsets of dimensions of interest. However, while ranking the features based on quality metrics, redundancies between the features may be high within the promoted set of features. The metrics-driven approaches do not in general take into account the redundancy relationships between the features. In our approach, we integrate ranking metrics as well as redundancy detection techniques for selecting more informative features.

Strategies for redundancy removal have been discussed in many machine learning approaches. In [MMP02] similar features are grouped iteratively and part of the group is removed based on a pre-defined threshold. The result set contains the features that are considered representatives of the groups. A similar but supervised selection process is described in [YL03] and its follow-up work [YL04], where the redundancy removal within a homogeneous group also considers the relevance of features to the target features. In [PLD05], the authors discuss the combination of min-redundancy and max-relevance and also evaluate the proposed methodology with different datasets. All these

approaches are effective in finding descriptive features, but less interactive for hypothesis verification. In our approach we are building our system upon these strategies and offer user interactive exploration so that their hypothesis can be verified and confirmed through visual presentation and interactive analysis.

The subspace searching approach [TMF+12], the VHDR method [YWRH03] in XmdvTool [War94], and Visual-FSSEM [DB00] are systems very similar to the concept of our work. [TMF+12] allows the user to refine the subspace found by a heuristic search process. [YWRH03] visualizes the relationship between features in a hierarchy structure. [DB00] takes user input at each iteration of the sequential forward search. However, the search part of the workflow in [TMF+12] does not support user input that it can hardly be altered as needed. The reduction schema in [YWRH03] does not provide a descriptive subset of features by default and the analysts need to do the selection by trial and error. In our system, we focus more on guidance for the analysts so that they may perform the analysis with default options. In the meantime, the can drill down the local analysis if needed.

The system in [BvLBS11] is capable of identifying feature redundancy based on pairwise comparison; the filtering stage of this approach retains the center of the group. In our approach, we integrate multiple filtering strategies, including the center based approach. Additionally, we allow the user to manually select different representatives for each group. SmartStripes [MBD+11] is a visualization system designed for redundancy discovery. Both the global redundancies and the local redundancies are visualized and presented to the user. However, in order to effectively remove the redundancy, proper guidance must be integrated, such as ranking and filtering strategies. Another limitation of the system, as the author suggested, is the partitioning process, which may be burdened by the selection of reference features. We address these limitations in our approach. Additionally, the user is able to use our system to identify alternative subsets of features after

spotting the local partitions of interest.

VaR [YPH+04] and DimStiller [IMI+10b] also inspired our work. The VaR system uses an MDS layout to show the similarities between features. Thus redundancies can be readily perceived. FeaVis provides more flexible interactive exploration methods in that similar features to a specified feature can be ranked. The correlation view in the DimStiller system removes the redundancies by a predefined threshold, and generates synthetic features to represent the removed ones. The goal of our work is to instead search for descriptive features that are readily communicable for different analysts. In that case, the derived mathematical representations are more difficult to understand.

## 3.7 Summary

We have described a hybrid system for feature selection that allows interaction and refinement at different level of details. It supports redundancy removal based on grouping and ranking features with default options. The analysts may choose to dive deep into one or several of the tasks such as local redundancy analysis.

Another contribution of this work is that our system integrates several commonly used quality metrics for filtering the features. Moreover, we also detect redundant features while analyzing the feature relationships. Based on this redundancy discovery process, we show the user the automatically selected non-redundant data features. Also, we allow the user to identify alternative non-redundant features according to their domain knowledge and visual feedback.

Lastly, our system enables local redundancy analysis in a three stage framework. The first stage shows the groups of features, where the user is able to identify the group of interest. The second stage shows group details, where the user is able to identify features of interest. Also, she is able to select or unselect any features to form an improved de-

scriptive subspace. The third stage shows patterns of local data partitions. The user can discover partitions of interest based on the visual feedback. She can also fine tune the selection of features for a subset of the data points.

# Chapter 4

# Local Model Diagnosis

A visualization system are demonstrated in this chapter for better *visual model diagnosis* where linear model training is embedded within visual interactions to facilitate model refinement. Most metrics for evaluating regression models are global in nature, and thus not useful for identifying local patterns. In this work, an integrated framework with visual representations is presented that allows the user to incrementally build and verify models to support local pattern discovery and summarization. This work enables the discovery of complementary models in terms of their performance locally on different subsets of the whole data space. A diversity measure is also provided to support the isolation of local models to reveal confounding factors during the regression analysis. Furthermore, this work integrates a hierarchical representation to identify abnormal local trends as well as common local trends. The former trend shares little with others while the later shares common characteristics such as slope and intercept. Real-world data is also used to evaluate the work and it shows the work is able to complement the computational algorithms in Weka. This part of work is published in EuroVis [ZWRH14].

In this chapter, three components are used to visually evaluate the performance of linear models from different perspectives. First, the *Model Complementarity* model eval-

uates the goodness of several feature combinations and measures how well they complement each other. In this view space (discussed in detail in Section 4.1), the model comparisons (Figure 4.3c) are visualized and presented to the analyst. This work also describes how to characterize the degree of complementarity between different features.

Second, the *Model Diversity* measures how much local diversity a set of features have. In this view space (discussed in Section 4.2), the local data spaces are generated via partitioning methods that are used to evaluate local performance of models in each partition. The measurement of diversity is also discussed in this space, which is ranked and visualized in Figures 4.3d&e.

Third, *Model Representivity* measures the degree of similarities between local models. In this view space (Section 4.3), we discuss how the representativity of a group of local models is measured. This helps us to determine how well a group of local models is represented by a single model. We also discuss how the view (Figure 4.3f) is designed to seek balance between coverage of a group of local models and the divergence within the group.

The overall workflow of the system is analogous to how a linear model is generated automatically. The first step is to evaluate which features are more relevant to the target feature. Then for the diversity metric, it measures how diverse the local performance of a selected feature set can be. Lastly, to avoid over-fitting, the local models can be merged after calculating the representativeness of them by clustering the model parameters.

To achieve the above goal, three model specific metrics are proposed and used in this work based on how much data are used to evaluate the quality (global measure vs. local measure) and how much data are used for constructing the models (global model vs. local model) as the Table 4.1 shows:

In the first space (top-left), linear models are built on *all data points* and the performance (goodness of fit) of the models are measured on *all data points* using Coefficient

| | **Global Measure** | **Local Measure** |
|---|---|---|
| **Global Model** | $R^2$, *RMSE* | Model Complementarity |
| **Local Model** | Model Representivity | Model Diversity |

Table 4.1: Model specific metrics for quality evaluation

of Determination ($R^2$) and Root Mean Squared Error (*RMSE*). This space together with 3 other spaces are shown in Table 4.1. Here, the *local measure* means the models are evaluated against a subset of data points. For example, companies with asset value below 1 million (small companies) and companies with asset value over 10 billions (large companies) can be two subsets of data in a financial dataset. The local models are the models specifically built in a local data space, such as a risk prediction model for small companies and another are for large companies. Since the metrics in this metric space has already been commonly studies by many other researchers, the metric spaces we primarily focused on in this work are model complementarity, model diversity and model representativity.

Below are two examples that show the insight of investigating local measures:

First, Figure 4.1) shows two models with bias towards opposite directions for part of the data space. Additionally, data with more complex structures can be described after tunning the simple linear models. These examples show that different local measures reveal different properties of the original model.

Regarding the this example, analysts may want to learn how the models complement each other locally, namely, *(a) on which subset of the data does one model have a smaller error than the other?* and *(b) on which parts of the data does one model overestimate the dependent variable while the other underestimates it?*

Second, Figure 4.2 shows different ways of defining multiple local models for the

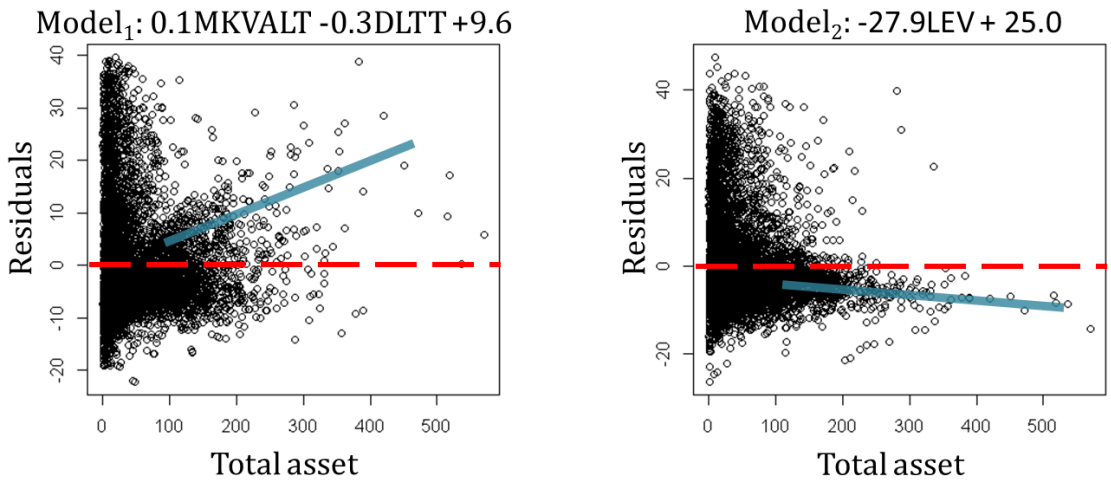


Figure 4.1: The two plots show that the two models displayed by the line trend oppose each other in terms of bias. $Model_1$ has the tendency to underestimate and $Model_2$ tends to overestimate when the total asset grows. The y-axis shows the goodness of fit (residuals). The x-axis is the value of total assets (one of the independent variables). DLTT: Total long-term debt; LEV: Leverage; MKVALT: Market value

same data. With different segments, the simple linear model can more flexibly represent the underlying data.

For this example, an analyst may want to understand *(a) are there any local models that significantly overperform the global model in terms of model fitness? (b) how many distinguishable local models are appropriate to describe the multiple trends in the data? (c) what are the best cutting values for forming appropriate subsets for isolating the local models?*

Two example solutions are: 1) to build local models on every single data point or; 2) to build one model for all the data points. However, the first case is overly complicated while the second case is not capable of capturing local patterns. This work focus on finding solutions inbetween these two. Regarding the isolated local patterns in example 2, an analyst may further ask, *(a) how different are these local models w.r.t. their direction (e.g., slope and intercept)? (b) do these local models comply with the direction of a representative global trend? (c) are there any outlier trends to oppose the majority of*

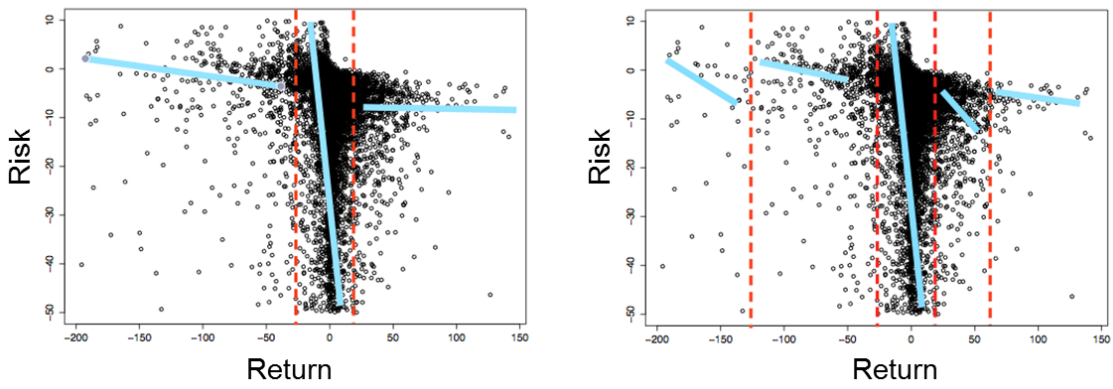


Figure 4.2: The plots represent how the linear relationship between two variables can differ when considering different partitions of data points. From a domain expert point of view, both high return and low return companies have relatively high risk; intermediate return (fluctuate around 0) companies tend to follow a trend whose risk is reversely proportional to the return.

*other trends?*. We will be discussing how we answer these questions in the following sections.

## 4.1 Model Complementarity Visualization

This section introduces: 1) how we measure goodness of fit of a model locally; 2) how we compare models based on their local measures; and 3) how we visualize the model complementarity based on the model comparison.

### 4.1.1 Goodness Measure

Consider the following scenario: ***A financial analyst found that a risk model she built is dominated by large companies. This means that the fitness (measured by residuals) is smaller for large companies. She wants to find out what additional variables can help the model to perform better on smaller companies.***

To make the scenario more specific, the dependent variable she uses is the bankruptcy

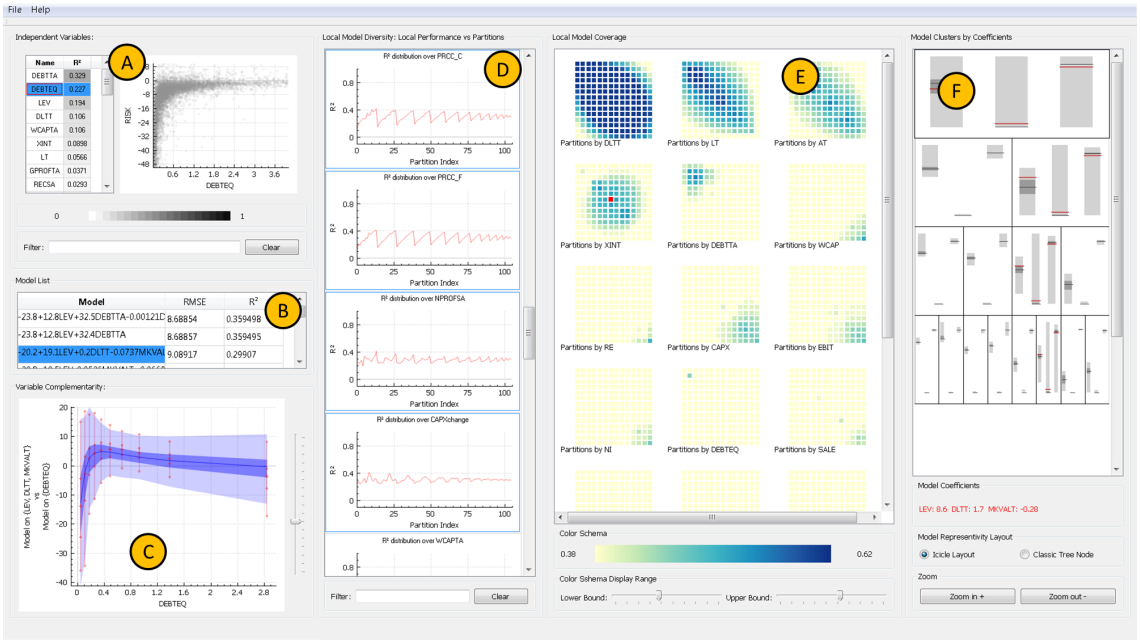


Figure 4.3: Integrated analysis framework with 3 stages. **1)** Variables are ranked by their relevance to the dependent variable. The scatterplot (a) shows the relationship between a selected independent variable and the dependent variable. The global models built by the analysts are listed in (b). Model complementarity is presented in (c) for refining a model in (b). **2)** Local models can be derived from a selected global model and are presented in (d,e). **3)** The local models are grouped and summarized in a hierarchy (f).

risk of companies labeled by financial analysts [WGG10]. The independent variables are financial attributes, such as working capital (WCAPTA), liability (DEBTTA and DEBTEQ), and total assets (AT). Next, the residual is defined as $Y - \hat{Y}$, where $Y$ is the dependent variable and $\hat{Y}$ is the predicted value. The analyst wants to learn for which portions of the data the model performs poorly, and for which portions of the data the model overestimates or underestimates. Hence, we need to investigate the local performance in local data subspaces using additional independent variables such as *total assets* to investigate whether there exists local models that behave differently from the globally trained model. The relationship between residuals of a linear model and the additional independent variable can illustrate where the model performs poorly (the small companies in this scenario).

### 4.1.2 Point-wise Comparison

Now, we conduct a point-wise model comparison. In Figure 4.1, the residuals of two linear models are plotted against an additional independent variable, *total assets*. Both models predict rather poorly as indicated by the large absolute values of residuals for the smaller companies. That is $model_1$ tends to under-estimate (positive residuals) the risk of larger companies while $model_2$ tends to over-estimate (negative residuals). In order to reduce such errors, we could take the following actions.

In practice, the two conditions for complementarity are: 1) *error complement*; 2) *bias complement*, as we will explain next. For a list of local partitions $p_1, p_2, \ldots, p_n$ of the given dataset D, let the local errors of a model $A$ be $e_1^a, e_2^a, \ldots, e_n^a$. The above two conditions for complementarity between models $A$ and $B$ are defined as:

$$\begin{aligned} &\exists i : (|e_i^a| >> 0 \Rightarrow |e_i^b| \to 0) \\ &\quad \vee (|e_i^b| >> 0 \Rightarrow |e_i^a| \to 0) \quad (i \in \mathbb{N}, i \leq n) \end{aligned} \tag{4.1}$$

$$\exists i : (e_i^a \approx \varepsilon \Rightarrow e_i^b \approx -\varepsilon) \quad (\varepsilon \in \mathbb{R}) \tag{4.2}$$

Intuitively, the two equations can be interpreted as: 1) the large errors of one model align with the small errors of another (Equation 4.1); 2) the over-estimation of one model aligns with the under-estimation portion of another (Equation 4.2).

### 4.1.3 Stacked Binned Summary View

A point-wise comparison becomes impractical as the number of data points gets larger. To help analyze the complementarity between models, we design a stacked binned summary view. The design is inspired by the visualizations for model local performance in [MP13],

where the residuals of two models are compared in a 2-D space-filling display using $|Y - \hat{Y}_1| - |Y - \hat{Y}_2|$ which is the performance difference of two models. Rather than showing the model differences we are instead interested in determining whether the combination of the two models is *cost-effective*. The cost is that adding each variable to a to-be-refined model increases the model complexity while helping with performance. Hence we want to know which variable helps to improve performance better.

We believe the models that complement each other form a better *combined model* (union of variables). The performance of the combined models can be examined in our table (see in Figure 4.3b). In order to compare the local performance of two models, we use Tukey's 5-number summary [Tuk77] to measure the distribution of residuals. In order to compare the local performance of two models, we need to plot two groups of box plots side by side. To visually enhance the comparison we provide a visual design (Figure 4.3c), to present the comparison and contrast. This particular design decision is made after experimenting with parallel bar charts and parallel box plots. The parallel bar charts only show the number of data points that fall into a particular partition, which is quite limited in determining the complementarity relationship. The parallel box plots provide more information but take a lot of screen space. Finally, we chose vertical lines with five dots as an alternative representations of classic box plots. To enhance comparison, we also use horizontal line connections to represent the second group of box plots by connecting the corresponding dots on each box plot.

Now we discuss how to define the local measures. A data partition (or range query) is needed to evaluate models locally. To define the data partitions, we use a reference variable driven partitioning method [MBD+11]. We chose the decomposition strategy that allows comparisons across other variables because we need to compare models that are formed by multiple variables over each data partition.

Next, we describe *variable rankings* in our system. Variable ranking is utilized to

support model refinement (Figure 4.3a) by showing the user the most promising variables first. The ranking score between an independent variable and the dependent variable are measured based on local partitions of the independent variable. Specifically, $R^2$ is computed for each partition formed by the independent variable of the dataset D. The final score then is the maximum $R^2$ over all these partitions.

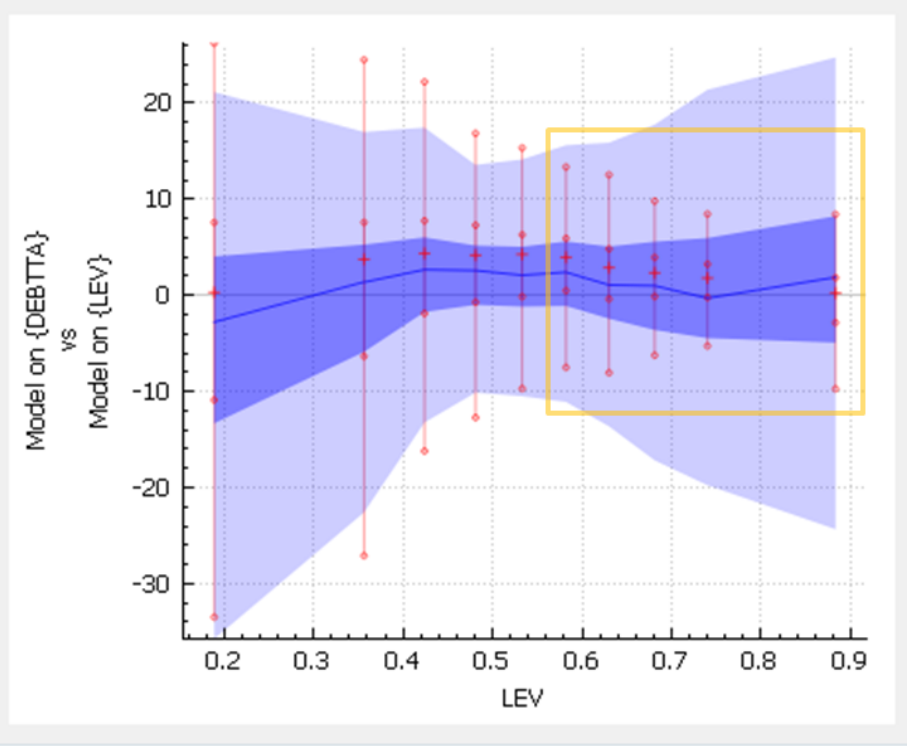


Figure 4.4: A candidate model LEV complements the to-be-refined model DEBTTA (in the yellow box). The y-axis represents the error spread of two models. Positive (Negative) values suggest bias towards underestimate (overestimate). The x-axis represents local partitions where the errors are estimated. The theme river design [HHN00] represents the residuals of the to-be-refined model. The red vertical lines represent the residuals of a candidate model (usually a univariate model).

With the model fitness comparison view designed in this space, the tasks a user can perform are listed as follows:

- *Identify relevant variables:* The users may freely choose a variable according to either its relevance to the dependent variable, or their previous domain knowledge.

- *Identify model weaknesses:* The visualization of model local measures reveals the distribution of residuals in local data spaces. By examining the local measures, a user may learn which parts of the data are not described effectively.

- *Identify complementary variables:* The visualization of local measures and local comparisons helps the user to identify whether adding variables to an existing model is cost-effective. The effectiveness of this strategy is evaluated in Section 4.4.2.

## 4.2 Model Diversity Visualization

This section discusses the problem when simply adding variables does not significantly improve the model fitness. According to previous work [MP13, GWR09], the reasons for this may be: 1) the trend is not linear, thus the refinement process must consider non-linear polynomials to be effective [MP13]; 2) there are multiple linear trends [GWR09]. In this work, we mainly focus on a domain-driven model coverage problem, namely, to seek a way for isolating multiple models and to label the trends with range queries so that local models can be associated with actual domain meanings. A query that contain a local pattern for example can be "*companies with income above 1 million*".

### 4.2.1 Isolating Multiple Local Models

***After an interactive selection process, the financial analyst is not satisfied with the model. She suspects there are multiple local trends in the dataset. Therefore she wants to break the dataset into a few partitions based on the size of the companies (total assets). Then, she builds local models for each partition.***

This task raises several interesting questions: 1) *how do we retain the domain meaning of each partition while we search for the local trends, and is this important?* 2) *how do*

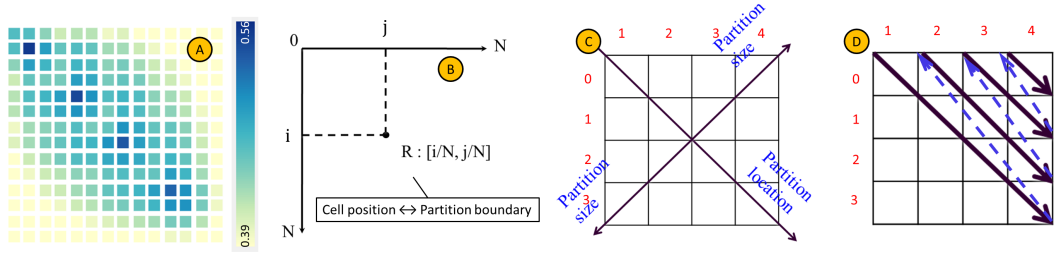


Figure 4.5: The x-y position of any cell in the grid view (a) is determined by the lower (x) and upper (y) percentile threshold of a data partition. The relationship between the x-y position and the partition boundary is shown in (b) and is indexed as in (c,d). Each cell is colored by the fitness of a local model in it. The diagonal and the orthogonal direction in (c) indicates two ways a data partition may change to another: expanding (add more data points) and shifting (add data points at one end and remove at the other). An time chart display (Fig 4.4b) of (a) is transformed from (a) by the sequence in (d) where the main diagonal is walked from top left first followed by the second diagonal above it. The walk continues till the right top corner.

*we define the partitions?* 3) *how do we illustrate the relationship between the possible ways of partitioning and the local trends each partition may have?*

For the first question, the analyst wants to isolate local based on different data partitions. She wants to know which companies (e.g., large companies or small companies) are associated with a particularly interesting local trend (Figure 4.2). To accomplish this task, we define a space $\mathscr{P} = \{p_1^1, p_2^1, \ldots; p_1^2, p_2^2, \ldots; \ldots; p_1^v, p_2^v, \ldots\}$ that contains partitions for $v$ variables where the $v$ variables are explored in the view we previously introduced. Once we have the partitions ready the next steps are to identify a linear trend in each partition using *Robust Regression* (as implemented in R [Hub11]), and visualize the model goodness (Figure 4.5a). The details about how to create partitions is discussed in Sec 4.2.2. The variables used in the local models are selected using the process discussed in Section 4.1.

In order to investigate how the trends are isolated into several data partitions. The first question to answer is whether local trends exist. Second, we need to annotate the local models with actual domain meanings. Then by linking a local trend to a range query such

as "large companies with more than 1 billion assets", the analysts are able to investigate the subset of data and further investigate the local properties of models.

### 4.2.2 Mutable Partitions

The discussions above lead to the second question. Specifically, *How do we assign the partition boundaries so that a trend is not divided into different partitions and irrelevant data points are minimized in a partition?* The question is also motivated by the representation of the piece-wise linear ranking model [MP13]: 1) when using very coarse piece-sizes, partitions are large and may contain irrelevant data points; 2) when using very fine segments, a trend may be assigned into several partitions. To address that, we use a enumerated partitioning strategy considering all interesting reference variables for partitioning and all interesting sub-intervals of partitions. For example, $total\ assets : [0/100, 30/100]$ represents a 0th and 30th percentile interval on reference variable *total assets*. Each partition in space $\mathscr{P}$ thus can be defined as $p_k^R = R : [l,h]$ where $R$ denotes the chosen reference variable; $k$ represents the index of the partition; and $l$ and $h$ ($0 \leq l,h \leq 1$) represent lower and upper boundaries on the reference variable. The space $\mathscr{P}$ is populated by partitions of varying boundaries, which is discussed next together with the layout strategy.

### 4.2.3 Partition Layout and Representation

We answer the third question by introducing the layout strategy of the diversity view (Figure 4.5a). In an $n$ by $n$ grid view (Figure 4.5a), the position $(i, j)$ of a cell (Figure 4.5b) represents the boundaries $[i/n, j/n]$ of a data partition. The factor $1/n$ is a *minimum step size threshold* to avoid infinite number of partitions. Due to the symmetricity of the $n$ by $n$ grid and the trivial information on the diagonal we first remove the diagonal and

the entries below the diagonal. Then we fill the lower half of the grid according to the symmetricity. We fill the grid because several test subjects felt the symmetric view is more pleasing while others have no preferences. In some cases a partition $R : [i/n, j/n]$ may not well cover a linear trend due to missing relevant data points or containing irrelevant data points. Alternative partitions $R : [(i+\varepsilon)/n, (j+\omega)/n]$ ($\varepsilon, \omega \in \mathbb{Z}$) need to be compared to $R : [i/n, j/n]$ for getting better boundary positions. A vicinity relationship between the compared partitions is depicted in Figure 4.5c in two directions to help the comparisons. The diagonal direction corresponds to partition shifting (i.e., $\varepsilon$ and $\omega$ changes towards the same direction). The anti-diagonal direction represents the expanding or shrinking of a partition. The color of each cell in Figure 4.5a represents the goodness of fit of the trend in that partition. We use relative measure $R^2$ to measure the goodness of fit because the absolute fitness measure, such as RMSE, is often driven by the value of the independent variables. This may cause unfair comparisons between data partitions.

To support the ranking and filtering of diversity views, we design a linear layout of the partitions (Figure 4.3d), that are ranked by the degree of fluctuation (Figure 4.6b,d). We use standard deviation of the local goodness of fit to quantify the fluctuations. The data partitions in a line chart (x-axis) are ordered by the diagonal walking sequence illustrated in Figure 4.5d. The more fluctuating line in Figure 4.6b indicates higher diversity. It suggests that the reference variable is effective in isolating multiple local trends. The smoother line in Figure 4.6d suggests the performance of isolated local models is similar to that of the global model. The diversity view is ordered and filtered using the same standard deviation measure.

A user can perform the following tasks, using the views designed in this space:

- *Identify reference variables:* With the local model diversity measure, a reference variable is ranked based on the fluctuation local model (Fig 4.6b). With the ranking metric, the user may identify variables that better isolate local models.

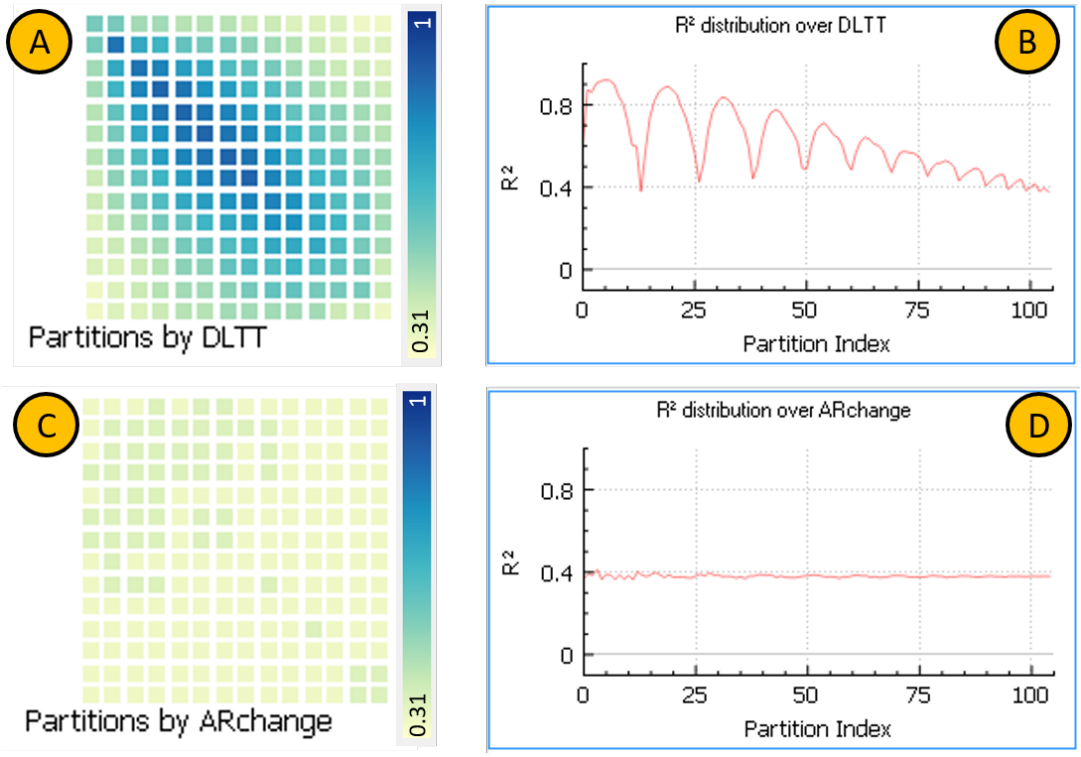


Figure 4.6: Visualize the degree of diversities. It shows that the local models isolated by partitioning on DLTT (a,b) have more diversity over the local models isolated by partitioning on ARChange (c,d). ARChange: Account Receivable Change

- *Identify multiple trends:* With the diversity representations, the user may identify multiple trends by reading the color spread in the diversity view (Fig 4.6a&c).

- *Identify the size, location and strength of a local trend:* The user may identify the corresponding range query for a trend in the diversity view by reading the x-y position of the cells. The size and strength of the trend can also be identified by the color spread the cells (Fig 4.6a&c).

## 4.3 Model Representativity Visualization

Let us continue our case scenario from Section 4.2. ***The financial analyst discovered that the local models perform rather well in some partitions*** ($profit$ : $[0.3, 0.5]$, $assets$ :

$[0.4, 0.7]$, *sales* : $[0, 0.4]$). ***She would like to confirm or rule out if these suggest the existence of a single model that can cover these local models. Furthermore, she also wants to know if that single model is robust, namely, are the local models it covers significantly diverging? Additionally, which data partitions contain trends that disagree with the majority of trends?***

### 4.3.1 Representative Trend

To help her, we designed an interactive hierarchical visualization that represents the similarities between the isolated models. We measure the similarities using coefficient vectors of the models (e.g., slope and intercept in a 2-D case). We want to answer: 1) *do the isolated local trends point to a similar direction, and thus can be covered by a representative trend?* 2) *if yes, how much confidence can be assigned to such local trends?* 3) *if not, how different are the trends in terms of their directions in the hyperspace?*

A representative model in $S$ (the set of local models over a selection of partitions) is expected to be central and cover as many partitions in $P$ (local partitions) as possible, while the divergence in $S$ is below a certain threshold $\xi$. We define $S$ as:

$$\min_{\forall S \subset P} (|P| - |S|) \text{ subject to } Div(S) < \xi$$

where $Div(S)$ denotes the model divergence in $S$ where $S$ is a group of partitions. To measure the model divergence, we use a normalized version of Euclidean distance:

$$d_{ij} = \sqrt{\frac{1}{w_a}(a_i - a_j)^2 + \frac{1}{w_b}(b_i - b_j)^2 + \ldots}$$

where $d_{ij}$ is the distance between two models $m_i$ and $m_j$ and $a_i, b_i, \ldots$ and $a_j, b_j, \ldots$ are the coefficients for the two models. The normalization factor we use is the amplitude of each coefficient: $w_a = max_i(|a_i|)$, $w_b = max_i(|b_i|)$, and so on. To visualize the divergence

and the coverage problem, we leverage the idea of *below traversal* in the hierarchical aggregation [EF10]. The idea is to cluster the local models based on model coefficients so that similar local models within one cluster may be represented by a more general model. The details are discussed in Sec 4.3.2.

### 4.3.2 Interactive Local Trend Aggregation

To support interactive local model clustering and understand the model coverage problem. We employ a divisive clustering algorithm [KR09] that divides a large cluster of items into smaller clusters in a top-down process. At each iteration it separates clusters of items at a computed cutting location. Icicle plots [KL83] are used to represent the hierarchical group structures. The icicle plots use relative positions of the node instead of binary representation of edges to infer parents and children thus it is believed to have higher information density than classic tree node graph [MR10]. The model divergence of each cluster is visualized at each node of the icicle plot using a variation of box-plot (Figure 4.7 right) where bars represent the coefficient statistics of the models. Using the techniques above, the representivity of a model $M_R$ in the partition space $S$ (a cluster of partitions) can be implied from the divergence of the models in $S$, the centrality of model $M_R$ in $S$ and the coverage of $S$. The divergence of models represents the degree of differences between model coefficients. The model divergence can be directly read from the box-plot in each node of the icicle plot where higher bars represent higher degree of divergence and lower bars represent low divergence. The centrality of a particular model can be discovered by linked interaction between the two views in Fig 4.7. Specifically, mouse over the color grids triggers a highlighted bar in the icicle plot. In Fig 4.8, the model at the edge of the heatmap view (red rectangle) has quite different model coefficients from its peer models as indicated by the horizontal red bars (appears at mostly very top or very bottom of the box plot) in the icicle plot. This example shows the model has very low centrality and it

cannot be used as a representative model.

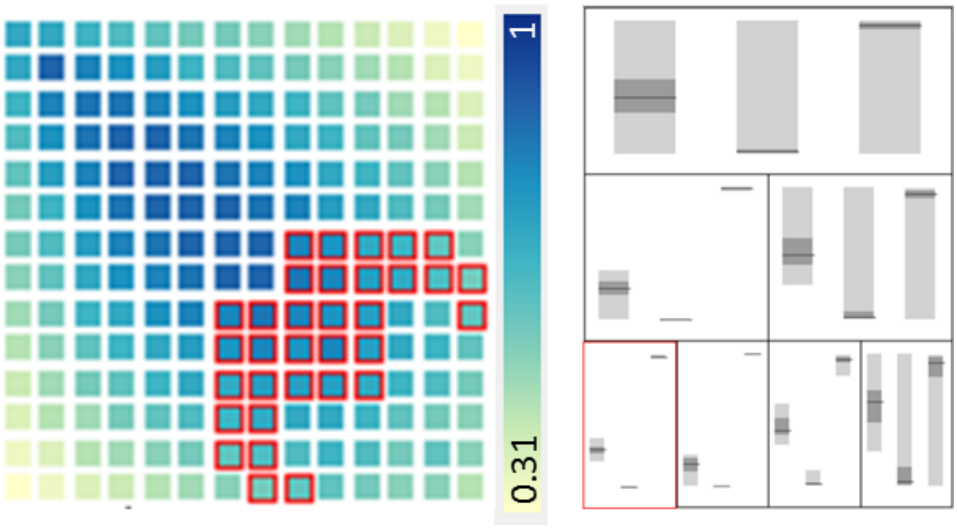


Figure 4.7: Visualize the coverage (cells with red outline on the left) of a selected cluster of data partitions (selected node marked with red rectangle on the right).

### 4.3.3 Aggregation Quality Loss

The user can double click on a node to break down a cluster with high divergence or merge smaller clusters with low divergence. The user may find the divergence of a cluster reduces to small values while still covering a set of data partitions (highlighted by red rectangles) (Figure 4.7). Also that highlighted cluster of local models is shown as the bottom left node in the icicle plot where the divergence of model parameters is low. As briefly discussed earlier. The user can also mouse over the heatmap view (Figure 4.8 left) and examine the centrality of the highlighted partition in a group (Figure 4.8 right). In this example, it is an outlier trend in the 2nd node at level 3 of the icicle plot (node with red bars in it) because all the three bars are at the boundary of the box-plot (Figure 4.8 right).

Additionally, the divergence of the group is higher than the other three groups at the same level. Another example can be seen in Figure 4.111 where the divergence of the

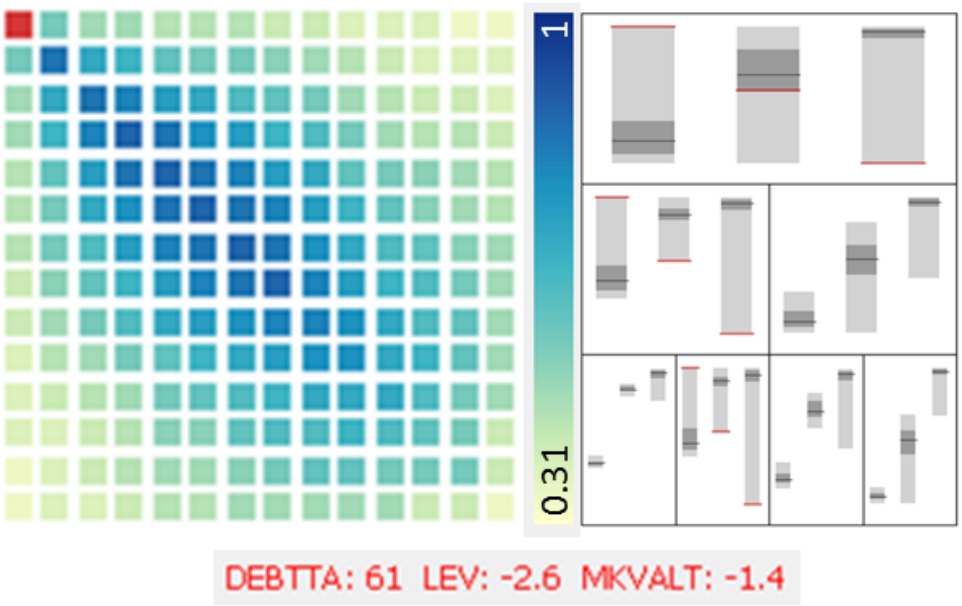


Figure 4.8: Visualize the coefficient vector (red horizontal bars in the icicle plot) of the linear trend in the highlighted data partition (left). The red text shows the value of the coefficients and the name of variables. The color scale shows the relative goodness of local models in a corresponding partition.

grouped model is lower than that in the previous example and the coefficients of the highlighted model are close to the center of the box-plot (Figure 4.11l). Lastly, the user may want to click on the nodes in the icicle plot (Figure 4.7 right) and examine the data coverage of each node (Figure 4.7 left) which is a linked interaction that highlights all corresponding partitions in the heatmap (Figure 4.7 left) corresponding to clicks on the nodes of the icicle plot.

This view space supports:

- *Identify outlier trends:* Coefficient values of a trend that are boundary values comparing to other trends may indicate that it is an outlier trend.

- *Identify a representative trend:* A representative trend can be identified by checking the divergence of the group it belongs to, centrality of the trends in the group and data coverage of the group.

## 4.4 Evaluation

In this section, we demonstrate a case study using a financial database. We also report the result of a user study we conducted involving professors and students from the departments of Math, Computer Sciences, and School of Business.

### 4.4.1 Case Study: Linear Models of Bankruptcy Risks

The data we use in this work are from Compustat [Poo11], a database of financial, statistical and market information of companies from around the world. Since the database is very large for visual analytics. It has more than 10 GB data collected for over 60 years. we focus on only on one sector of the US companies that are active in the year 2010, namely the service sector classified by the SIC standard [sic13]. After this cleaning, we acquired 45 variables suggested by domain expert for 9,483 observed companies that are in the selected service sector.

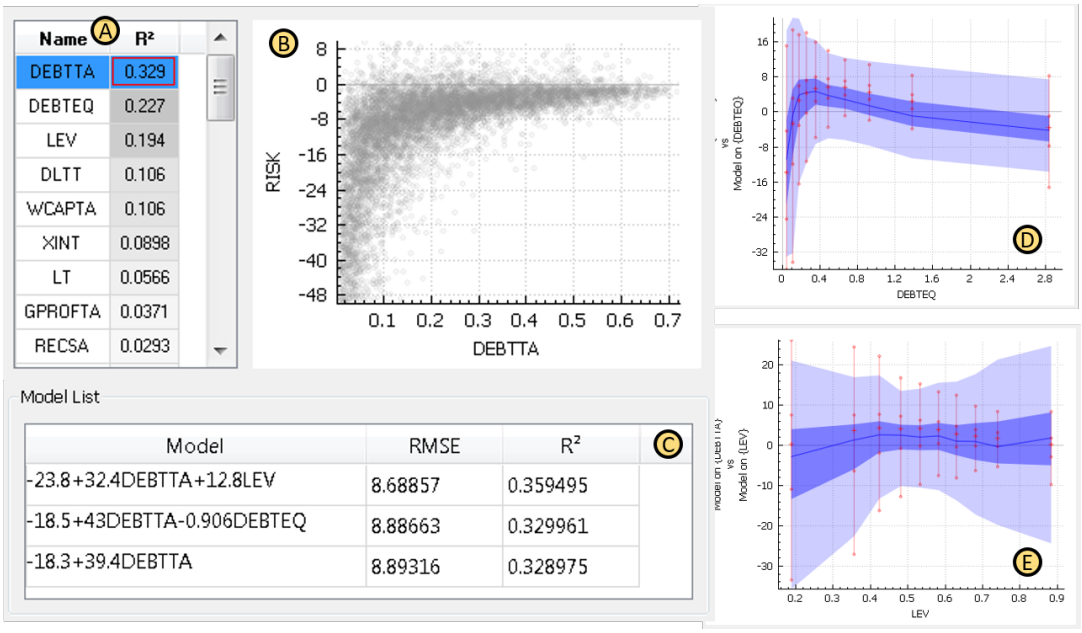


Figure 4.9: A case study for modeling risk. a) A ranking list of independent variables. b) Scatterplot of a selected independent variable and the dependent variable. c) A list of built models. d, e) Complementarity analysis.

To build linear models for risk prediction, the analyst first examines the relevance ranking scores of the independent variables in the relevance view by computing the dependency between all the independent variables and the dependent variable (Figure 4.9a). The relationship between the highlighted independent variable and the dependent variable can also be plotted in a scatterplot (Figure 4.9b) to examine the relationship in detail. From the relevance ranking list, she identifies that the variables DEBTTA, DEBTEQ, and LEV are most predictive for the dependent variable. However, she would like to figure out which combination is better. Choosing all 3 of them is an option, but it may increase the model complexity unnecessarily.

She next examines the model complementarity view (Figures 4.9d and 4.9e) to determine which variable *complements* the variable DEBTTA (the first candidate) better. The two models in Figure 4.9d share a common pattern (up/down and vertical spread, and less complementary). The model represented as red lines in Figure 4.9e performs better at the right half of the data partitions (smaller error spreading, and more complementary). She confirms that the combination $\{DEBTTA, LEV\}$ is better ($RMSE = 8.68, R^2 = 0.359$) than $\{DEBTTA, DEBTEQ\}$ ($RMSE = 8.89, R^2 = 0.330$) in the model list (Figure 4.9c) after trying both combinations. Although both of them are better than model with only one variable $\{DEBTTA\}$ ($RMSE = 8.89, R^2 = 0.329$), $LEV$ is the variable that adds more fit. In an automatic model building process, the analyst would not have had direct control over this variable selection, the expert knowledge thus cannot be directly applied to help the selection.

Next, the analyst may examine the local models that are derived from the current best model. The derived local models are based on the same set of variables we identified via the *complementarity analysis*. Each local model is built on a partition ($R : [l, h]$). By examining the *model diversity* views, the analyst immediately notices two interesting patterns: 1) Figure 4.10f shows that in some partitions (in Figure 4.10g cells with darker

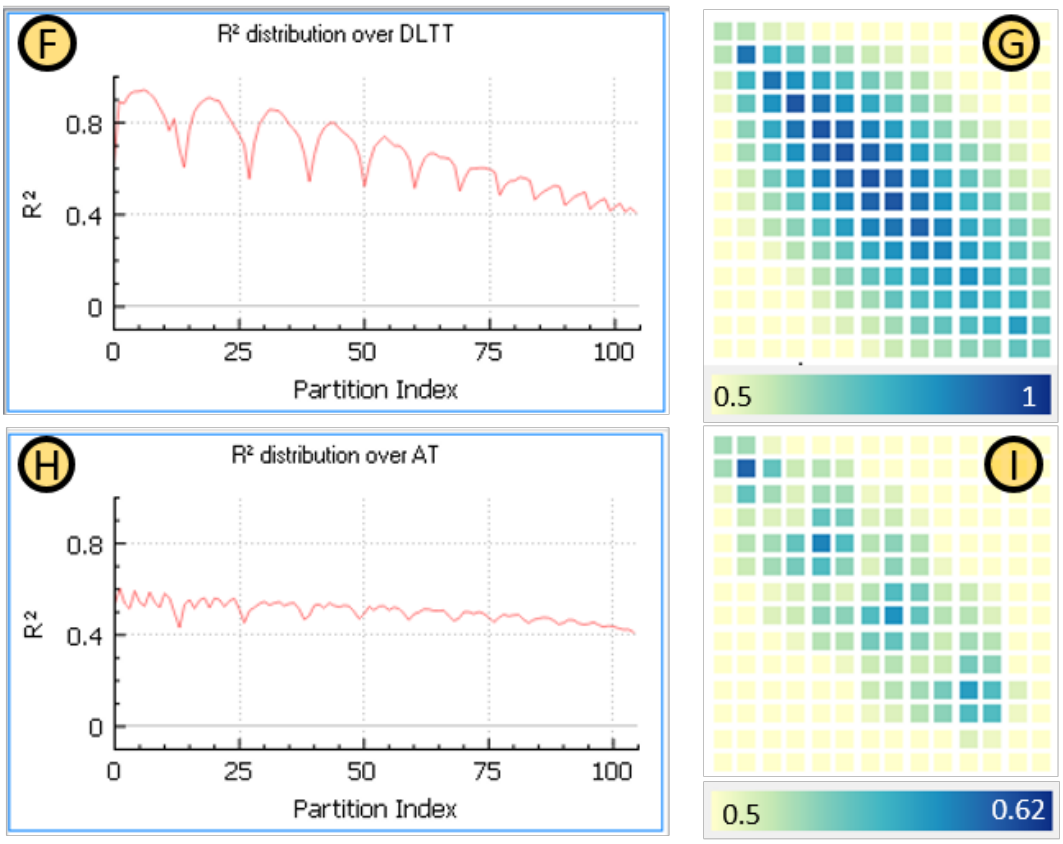


Figure 4.10: A case study for modeling risk. f), g), h) and i) Local model diversity analysis.

blue), the local trends are very strong, as $R^2$ is over 0.9 in some of them. The strong linear trends can be expanded along the orthogonal direction (Figure 4.10g) to a larger range of partitions at a lower threshold (lighter colors). 2) Another pattern that could be spotted is that the local models show 4 local maxima in Figure 4.10i, where 4 strong linear trends are isolated in the partitions represented by the darker blue cells. The pattern shows that the domain knowledge of the analyst is partially correct in the sense that the local trends are indeed stronger when isolating them by the variable *total assets*.

It suggests that constructing models with a mixture of both small and large companies is less effective because the model with only smaller companies (the dark cell at $R : [1/14, 2/14]$ in Figure 4.10i) outperforms the model built on all companies (top-right cell

at $R : [0/14, 14/14]$ in Figure 4.10i). The reason she is only partially correct is that the 4 local maxima in Figure 4.10i suggest modeling the companies at 4 different scales instead of 2.

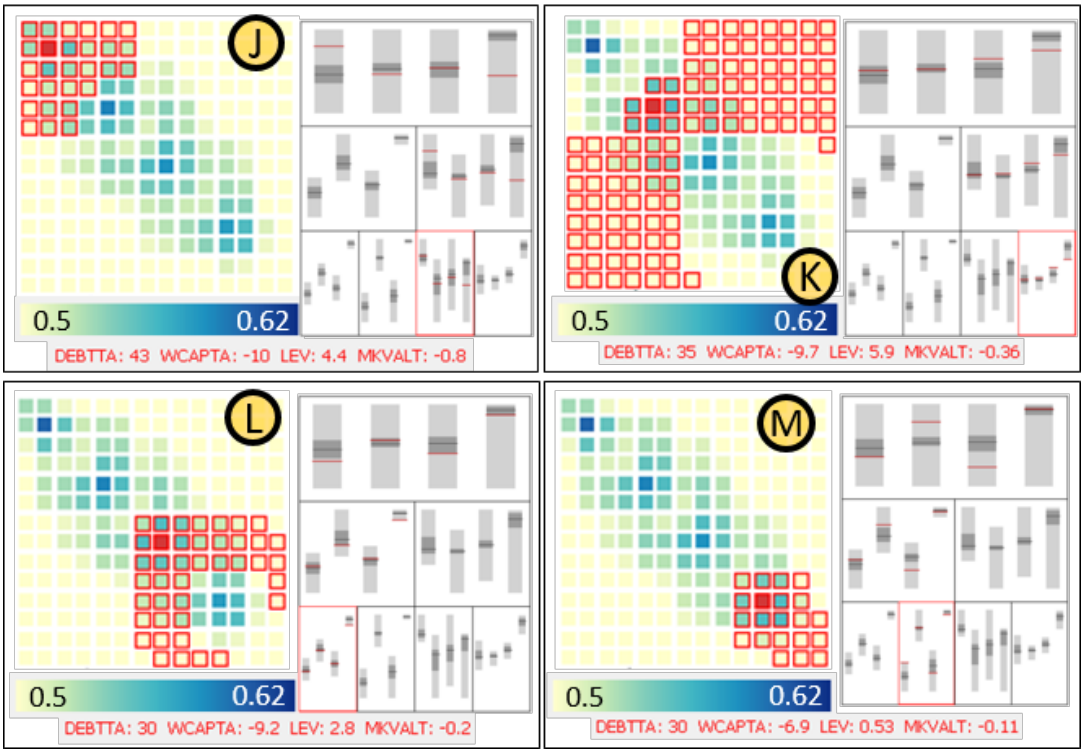


Figure 4.11: A case study for modeling risk. j), k), l) and m) Model representivity analysis.

The next step is to check the *model representivity*. The analyst breaks the local models down hierarchically, and discovers that at level 3 each of the 4 clusters contains one local maximum (Figure 4.11j, 4.11k, 4.11l, 4.11m). This confirms that using the group of 4 is the right choice, because the directions of the trends in the 4 clusters are different. Specifically, *DEBTTA* and *MKVALT* are more significant in the small company group and the significance decreases with the scale of the companies. *WCAPTA* and *LEV* are less significant in the large medium and large groups, while *WCAPTA* is most significant in the small medium group. Another notable pattern is that the local trend in the small medium group can be represented by a more general trend, because the coefficients of the

trends in that group has rather small variances indicated by the heights of the bars in the icicle plot as show in Fig 4.11k.

The three model spaces in this part of work are additional features that complement the automatic model building process. We compare our approach to the *LinearRegression* algorithm in Weka from the perspective of model complexity (number of variables) and model fit ($R^2$). Using the same dataset as input, Weka selects 27 out of the original 45 variables and forms a linear model with $R^2$ at 0.522. This overall fit in the whole data space is better than the models we formed in LoVis which usually involve much fewer variables (4 or 5). However, LoVis has the advantage of modeling the local properties of the dataset. 1) It discovers local data spaces that can form linear models with $R^2$ at above 0.8 (Figure 4.10f,g) which is higher than the fit of the automatically formed global model; 2) It also characterizes multiple local models with local maximal fit (Figure 4.10h,i). With only 4 variables, each model has $R^2$ of about 0.6 which is higher than the fit of the automatically formed model on 27 variables.

### 4.4.2 User Study for Evaluating Model Fit

To validate the usability of the model complementarity metric and the corresponding views, we performed a user study with 20 subjects. The participants answered 3 questions after a short training. In each question, they were asked to choose one option out of two. The ground truth is that one option that shows a model formed based on a set of variables (e.g. Figure 4.9e) is better than the other (e.g. Figure 4.9d) that is formed by a different set of variables. We expected to see the user selected option is better by examining the complementarity view to compare two models. Specifically the difference of two models are measured using FD score that is defined as below:

$$FD = |\text{Model Fit}_{\text{variable set 1}} - \text{Model Fit}_{\text{variable set 2}}|$$

Table 4.2: User study accuracy results based on 3 questions.

| FD ($R^2$) | Accurary (%) | Avg time(s) |
|---|---|---|
| 0.12 | 90 | 13.4 |
| 0.08 | 80 | 24.6 |
| 0.03 | 60 | 25.3 |

The FD score is calculated by using the model performance of one set of variables minus the other. In the results, there is a relationship between the *selection accuracy* and the FD scores between the two options, as show in Table 4.2.

From the result, more users (90%) made optimal selections when the FD between the two choices is more significant (0.12). Here we define accuracy as the percentage of subjects who made the right choice. When the FD goes down to 0.03 ($R^2$), the user selection tends to be less accurate (60%) and is more time consuming (25.3$s$). However, at that point, the performance gain of adding the wrong selection is only 0.03 (measured by $R^2$) less than the right selection. Based on this user study we shown that the metric and visual design of this work is useful to guide the user make right choices most of the time, it is less useful when the two choices leads to very similar models in terms of model fitness.

## 4.5 Related Work

Many methods for identifying local patterns exist. Guo et al. [GWR09] proposed a system to isolate linear trends by only including the data points within a user specified distance to a trend. Their idea of isolating multiple trends is similar to ours, except that our methods use partition-driven methods to describe the meaning of multiple linear trends. The local patterns in paper [GWRR11] are defined around a focal point; the relative positions of neighbouring points of it are visualized. In LoVis, however, we are instead interested in the local pattern of *a group* of data points and the comparisons between groups.

A partition-based framework [MP13] compares the linear models in both 1-D and 2-D partitions of independent variables to facilitate variable selection. In LoVis, we are more interested in how the variables locally complement each other, how the performance of local models varies in different data partitions, and how to identify the representativeness of local patterns. A maximal information coefficient (MIC) metric [RRF+11] was defined for identifying multiple types of pair-wise relationships via local analysis. In LoVis, we focus on one type of local relationship and investigate the local pattern of models formed by *multiple variables*.

Data partitioning is perhaps the most important step for identifying local patterns; an interactive framework [MBD+11] was implemented to guide the user to identify local relevance and aggregated global correlation. We do not intend to solve the problem of searching locally correlated feature sets and the corresponding subset of data points, which leads to an expensive optimization problem [GFVS12]. In our work, we instead leverage the knowledge of analysts to make choices to reduce the search space by partitioning on variables of interest that show fluctuations regarding local model performance.

The Rank-by-Feature Framework [SS04] is similar to our work; it provides quality metrics to measure the interestingness of lower projections (1-D and 2D) to facilitate the visual exploration process in high dimensional data. It has inspired our work in the sense of ranking views by importance. Models with diverse goodness of fit are believed to have more prediction power [BWHY05] and they may indicate the existence of a "lurking explanatory variable" [BHO+75]. Other techniques that focus on the application of quality measures are not specifically designed for local pattern discovery, though they indeed inspired us. Scagnostics [WAG05] proposed metrics for identifying interesting structures (e.g., clumpy and stringy). The user-centric approach [JJ09] utilizes several quality metrics that could be combined and adjusted by the user. Peng et al. [PWR04] proposed a metric for reducing clutters in the visual representations. Peringer et al. [PBH08]

suggested a quality measure integrated with data space brushing and linking. Tatu et al. [TMF+12] implemented a system that ranks data variables based on subspace cluster structures. The EnsembleMatrix [TLKT09] combines multiple model analysis with visual representations. It allows the user to visually examine the contrast of multiple classifiers and interactively combine them. This strategy motivated us to build a framework to investigate the relations between multiple models. Additionally, we allow the user to incrementally examine the model comparisons in terms of model complementarity and determine the best candidate models for combining.

## 4.6 Summary

In this work, we presented the LoVis system that integrates three visual spaces, focusing on local pattern discoveries that facilitate the linear model refinement process. We measure the degree of complementarity between a to-be refined model and the candidate variables so that a suitable variable can be selected to compensate for the poor performance of the to-be refined model locally. Local models are built to model the diversity in the dataset in a novel partition space. Divergence of the local models is measured and visualized to investigate the representivity of a group of models.

# Chapter 5

# Conclusion and Future Directions

## 5.1 Conclusion:

This dissertation contributed to three research direction of visual exploration by integrating machine learning techniques into a comprehensive interactive framework for local pattern discovery. The three areas are closely related to what data analysts do in their every day work, namely, data exploration, feature exploration and model diagnosis. Each of the three types of visual explorations are based on leveraging machine learning tasks that facilitate the data modeling and summarization. Unlike other work that combines machine learning and visualization to perform rather specific tasks such as facilitate decision tree building or optimizing clutter issues in a visual display, this dissertation aims to provide novel frameworks that unify modeling tasks as well as visualization techniques to make the general data exploration task more efficient and effective.

The main contributions of this dissertation are:

- A system (MaVis) integrated multi-model strategy for time series co-movement analysis. Which is a prevalent pattern discovery problem in various application domains such as finance, business, medical science and engineering. The co-

movement of a collection of time series are measured using multiple time series models and allow the analysts to compare and contrast with ease in an interactive framework. State-of-the-art visual analytic techniques usually use specific data mapping strategies for time series pattern discovery. Compared to these approaches, adopting time series models in this work allows analysts to choose the modeling techniques that most suitable to their tasks.

- A system (FeaVis) implemented multiple feature similarity metrics for feature relationship visual inspection. FeaVis supports automatic redundancy removal based on grouping and ranking data dimensions. It integrates commonly used feature similarity metrics for investigating feature relationships.

- A system (LoVis) provides a novel way of diagnosing regression models locally. The degree of complementarity between models is measured to facilitate model refinement. Local models can be identified to describe the diversity of the dataset for gaining insight. Divergence of the local models is also measured to help identify common patterns of the model fitness that locally presents.

## 5.2 Future Directions:

There are several interesting directions for future research based on this work.

First, we are mostly interested in local patterns for the three areas of work. However, it is still a quite challenging problem to show the overall landscape of the data while providing insights to certain local subsets of the data space that contain interesting information. It is however a very expensive computational approach [GFVS12] to search for local space patterns. To design visual systems to reduce the complexity of computational tasks is a promising direction that may benefit the whole data science field. For example, for any given machine learning tasks, it executes the predefined algorithms to run though

the possibly large dataset. It is quite often that a data scientist realize there is a problem in the run after spending a long time waiting for the process to complete. Combining with visual system can potentially relieve such problem by showing partial results that only involves models trained locally.

Second, we support user metrics based on predefined metrics to determine similarities between features. However, it is a challenging problem to scale the number of metrics in the process of combining multiple metrics to a user metric using a set of weights. We currently only support three metrics and to parameterize the weight combination which is computationally feasible however, the required caching can quickly grow up when the number of metrics increases or the granularities of weight adjustment gets finer. Solving this problem will be really helpful for interactive analysis in case of multiple metrics.

Third, industry deployment of this work will be useful for evaluation. The system designed in this dissertation work can be used and improved by collecting feedback from analysts while using this work in their daily work. The evaluation can be beneficial for the visual analytic field if we can understand what the bottlenecks are during the data analytics process. The evaluation is especially useful when an analyst has inadequate knowledge while performing an analytic task. In that case, the feedback we collect can be valuable to generate principles for visual designs to help analysts gain insights even when they work on a type of new dataset.

Fourth, the scalability of the systems are not designed for larger dataset mainly due to computational cost rather than visual rendering cost. Generating appropriate partitions before forming local models is an expensive search problem. Another potential contribution in this field is to reduce the cost of computations based on interactive human adjustment. In that case, a human expert may identify a non-promising searching strategy and terminate it early enough to make the search more effective. In case the computational process may take a long time, the intermediate feedback may help analysts to make decisions in a

timely manner.

Finally, this work integrates two types of models (i.e., regression model and time series model). It is still a challenging problem to integrate visual interactions to all the machine learning processes in general. It requires efforts from data scientists to design and tweak the machine learning algorithms to provide rich feedback for the analysts to make sense of the machine learning process.

# Bibliography


[ABHA09] Wafa Abdelmalek, Sana Ben Hamida, and Fathi Abid. Selecting the best forecasting-implied volatility model using genetic programming. *Advances in Decision Sciences*, 2009, 2009.

[ABK98] M. Ankerst, S. Berchtold, and D.A. Keim. Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 52–60, 1998.

[AEL+10] Georgia Albuquerque, Martin Eisemann, Dirk J Lehmann, Holger Theisel, and Marcus Magnor. Improving the visual analysis of high-dimensional datasets using quality measures. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 19–26, 2010.

[AHH+14] B Alsallakh, A Hanbury, H Hauser, S Miksch, and A Rauber. Visual methods for analyzing probabilistic classification data. *IEEE TVCG*, 20(12):1703–1712, 2014.

[Ale08] Carol Alexander. *Moving Average Models for Volatility and Correlation, and Covariance Matrices*. John Wiley & Sons, Inc., 2008.

[Alt12] Edward I Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4):589–609, 2012.

[AMST11] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, London, 2011.

[AWD12] Anushka Anand, Leland Wilkinson, and Tuan Nhon Dang. Visual pattern discovery using random projections. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 43–52, 2012.

[BB01] H Thomas Banks and Kathleen L Bihari. Modelling and estimating uncertainty in parameter estimation. *Inverse Problems*, 17(1):95, 2001.

[BHO+75] Peter J Bickel, Eugene A Hammel, J William OConnell, et al. Sex bias in graduate admissions: Data from berkeley. *Science*, 187(4175):398–404, 1975.

[BLBC12] Eli T Brown, Jingjing Liu, Carla E Brodley, and Remco Chang. Disfunction: Learning distance functions interactively. In *Visual Analytics Science and Technology (VAST)*, pages 83–92, 2012.

[Blo09] Nicholas Bloom. The impact of uncertainty shocks. *Econometrica*, 77(3):623–685, 2009.

[BM01] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250, 2001.

[Boc07] Hans-Hermann Bock. Clustering methods: a history of k-means algorithms. In *Selected contributions in data analysis and classification*, pages 161–172. Springer, 2007.

[BOL12] Ken Brodlie, Rodolfo Allendes Osorio, and Adriano Lopes. A review of uncertainty in data visualization. In *Expanding the Frontiers of Visual Analytics and Visualization*, pages 81–109. 2012.

[BPS14] Bruce A Blonigen, Jeremy Piger, and Nicholas Sly. Comovement in gdp trends and cycles among trading partners. *Journal of International Economics*, 94(2):239–247, 2014.

[BTK11] Enrico Bertini, Andrada Tatu, and Daniel Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.

[BTV14] Andrea Buraschi, Fabio Trojani, and Andrea Vedolin. When uncertainty blows in the orchard: Comovement and equilibrium volatility risk premia. *The Journal of Finance*, 69(1):101–137, 2014.

[BvLBS11] Sebastian Bremm, Tatiana von Landesberger, Jürgen Bernard, and Tobias Schreck. Assisted descriptor selection based on visual comparative data analysis. In *Computer Graphics Forum*, volume 30, pages 891–900, 2011.

[BWHY05] Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005.

[CC00] Trevor F Cox and Michael AA Cox. *Multidimensional scaling*. Chapman & Hall/CRC, 2000.

[CEG+09] Gayathri Chandrasekaran, Mesut Ali Ergin, Marco Gruteser, Richard P Martin, Jie Yang, and Yingying Chen. Decode: Exploiting shadow fading

to detect comoving wireless devices. *Mobile Computing, IEEE Transactions on*, 8(12):1663–1675, 2009.

[CL98] Robert E Carpenter and Daniel Levy. Seasonal cycles, business cycles, and the comovement of inventory investment and output. *Journal of Money, Credit and Banking*, pages 331–346, 1998.

[CWRY06] Qingguang Cui, Matthew O Ward, Elke A Rundensteiner, and Jing Yang. Measuring data abstraction quality in multiresolution visualizations. *IEEE TVCG*, 12(5):709–716, 2006.

[DB00] Jennifer G Dy and Carla E Brodley. Visualization and interactive feature selection for unsupervised data. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 360–364, 2000.

[DBKMR05] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

[EF10] Niklas Elmqvist and J-D Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):439–454, 2010.

[EpKSX96] Martin Ester, Hans peter Kriegel, Jrg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

[FFM12] Fabian Fischer, Johannes Fuchs, and Florian Mansmann. Clockmap: Enhancing circular treemaps with temporal glyphs for time-series data. *Proc. EuroVis Short Papers, Eurographics*, pages 97–101, 2012.

[FGP+13] Christoph Flamm, Andreas Graef, Susanne Pirker, Christoph Baumgartner, and Manfred Deistler. Influence analysis for high-dimensional time series with an application to epileptic seizure onset zone detection. *Journal of Neuroscience Methods*, 214(1):80–90, 2013.

[GB14] A.C. Guidoum and K. Boukhetala. *Sim.DiffProc: Simulation of Diffusion Processes.*, 2014. R package version 2.9.

[GFVS12] Stephan Günnemann, Ines Färber, Kittipat Virochsiri, and Thomas Seidl. Subspace correlation clustering: finding locally correlated dimensions in subspace projections of the data. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 352–360. ACM, 2012.

[GLG+13] Samuel Gratzl, Alexander Lex, Nils Gehlenborg, Hanspeter Pfister, and Marc Streit. Lineup: Visual analysis of multi-attribute rankings. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2277–2286, 2013.

[GNRM08] Supriya Garg, Julia Eunju Nam, IV Ramakrishnan, and Klaus Mueller. Model-driven visual analytics. In *Visual Analytics Science and Technology (VAST)*, pages 19–26, 2008.

[GWR09] Zhenyu Guo, Matthew O. Ward, and Elke A. Rundensteiner. Model space visualization for multivariate linear trend discovery. *IEEE Symposium on Visual Analytics Science and Technology*, pages 75–82, 2009.

[GWR11] Zhenyu Guo, Matthew O Ward, and Elke A Rundensteiner. Nugget browser: Visual subgroup mining and statistical significance discovery in multivariate datasets. In *Information Visualisation (IV), 2011 15th International Conference on*, pages 267–275, 2011.

[GWRR11] Zhenyu Guo, Matthew O. Ward, Elke A. Rundensteiner, and Carolina Ruiz. Pointwise local pattern exploration for sensitivity analysis. *IEEE Conference on Visual Analytics Science and Technology*, pages 129–138, 2011.

[HHN00] Susan Havre, Beth Hetzler, and Lucy Nowell. Themeriver: Visualizing theme changes over time. In *IEEE Symposium on Information Visualization*, pages 115–123. IEEE, 2000.

[HS04] Harry Hochheiser and Ben Shneiderman. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18, 2004.

[Hub11] Peter J Huber. *Robust statistics*. Springer, Berlin Heidelberg, 2011.

[IMI+10a] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. Dimstiller: Workflows for dimensional analysis and reduction. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 3 –10, 2010.

[IMI+10b] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner, and T. Möller. Dimstiller: Workflows for dimensional analysis and reduction. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 3 –10, 2010.

[JE12] Waqas Javed and Niklas Elmqvist. Exploring the design space of composite visualization. In *Pacific Visualization Symposium (PacificVis)*, pages 1–8, 2012.

[JJ09] Sara Johansson and Jimmy Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. volume 15, pages 993–1000. IEEE, 2009.

[JME10] Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. Graphical perception of multiple time series. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):927–934, 2010.

[KCHP93] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series: A survey and novel approach. In *Data mining in Time Series Databases. Published by World Scientific*, pages 1–22, 1993.

[Kin06] Robert Kincaid. Line graph explorer: scalable display of line graphs using focus+context. In *In Working Conference on Advanced Visual interfaces*, pages 404–411. ACM Press, 2006.

[KKL12] Mark J Kamstra, Lisa A Kramer, and Maurice D Levi. A careful reexamination of seasonality in international stock markets: Comment on sentiment and stock returns. *Journal of Banking & Finance*, 36(4):934–956, 2012.

[KL51] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[KL83] Joseph B Kruskal and James M Landwehr. Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168, 1983.

[kld] Symmetrizing the kullback-leibler distance. http://www.ece.rice.edu/~dhj/resistor.pdf. Accessed: 2015-12-10.

[KM99] Regina Kaiser and Agustin Maravall. Estimation of the business cycle: A modified hodrick-prescott filter. *Spanish Economic Review*, 1(2):175–206, 1999.

[KP08] Jarl Kallberg and Paolo Pasquariello. Time-series and cross-sectional excess comovement in stock indexes. *Journal of Empirical Finance*, 15(3):481–502, 2008.

[KR09] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[KSM07] Efstathios Kirkos, Charalambos Spathis, and Yannis Manolopoulos. Data mining techniques for the detection of fraudulent financial statements. *Expert Systems with Applications*, 32(4):995–1003, 2007.

[KSS04] Daniel A Keim, Jörn Schneidewind, and Mike Sips. Circleview: a new approach for visualizing time-related multidimensional data sets. In *Proceedings of the working conference on Advanced visual interfaces*, pages 179–182. ACM, 2004.

[LJH13] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430, 2013.

[LWW90] Jeffrey LeBlanc, Matthew O Ward, and Norman Wittels. Exploring n-dimensional databases. In *Proceedings of the IEEE Conference on Visualization*, pages 230–237, 1990.

[Mal89] Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, 1989.

[Mas08] Philippe Masset. Analysis of financial time-series using fourier and wavelet methods. *Available at SSRN 1289420*, 2008.

[MBD+11] Thorsten May, Andreas Bannach, James Davey, Tobias Ruppert, and Jörn Kohlhammer. Guiding feature subset selection with an interactive visualization. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 111–120, 2011.

[MHS07] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. Show me: Automatic presentation for visual analysis. *IEEE TVCG*, 13(6):1137–1144, 2007.

[MMP02] P. Mitra, C.A. Murthy, and S.K. Pal. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301–312, 2002.

[MP13] Thomas Muhlbacher and Harald Piringer. A partition-based framework for building and validating regression models. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962–1971, 2013.

[MR10] Michael J McGuffin and Jean-Marc Robert. Quantifying the space-efficiency of 2d graphical representations of trees. *Information Visualization*, 9(2):115–140, 2010.

[MUCM03] Rahim Moineddin, RE Upshur, Eric Crighton, and Muhammad Mamdani. Autoregression as a means of assessing the strength of seasonality in a time series. *Popul Health Metr*, 1(1):10, 2003.

[MZ08] AI McLeod and Ying Zhang. Faster arma maximum likelihood estimation. *Computational Statistics & Data Analysis*, 52(4):2166–2176, 2008.

[PBH08] Harald Piringer, Wolfgang Berger, and Helwig Hauser. Quantifying and comparing features in high-dimensional datasets. In *In Proceedings of the IEEE Symposium on Information Visualisation*, pages 240–245, 2008.

[PK10] Mark Pinsky and Samuel Karlin. *An introduction to stochastic modeling*. Academic press, Oxford, UK, 2010.

[PLD05] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[Poo11] Standard & Poor's. Compustat database. www.compustat.com, July, 2011. Accessed: 2013-11-27.

[PWB+09] Kristin Potter, Andrew Wilson, P-T Bremer, Dean Williams, Charles Doutriaux, Valerio Pascucci, and Chris R Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *Data Mining Workshops, ICDMW*, pages 233–240, 2009.

[PWR+] Wei Peng, Matthew O Ward, Elke Rundensteiner, et al. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 89–96. IEEE.

[PWR04] W. Peng, M.O. Ward, and E.A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 89–96, 2004.

[R C12] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. ISBN 3-900051-07-0.

[Ram02] James Bernard Ramsey. Moments and the shape of histograms. In James Bernard Ramsey, Joseph H. Newton, and Jane L. Harvill, editors, *The Elements of Statistics: With Applications to Economics and the Social Sciences*, chapter 4, pages 77–107. Duxbury/Thomson Learning, Belmont, CA, 2002.

[RRCZ14] Juan Carlos Reboredo, Miguel A Rivera-Castro, and Gilney F Zebende. Oil and us dollar exchange rate dependence: A detrended cross-correlation approach. *Energy Economics*, 42:132–139, 2014.

[RRF+11] David N Reshef, Yakir A Reshef, Hilary K Finucane, Sharon R Grossman, Gilean McVean, Peter J Turnbaugh, Eric S Lander, Michael Mitzenmacher, and Pardis C Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.

[SCL+12] Conglei Shi, Weiwei Cui, Shixia Liu, Panpan Xu, Wei Chen, and Huamin Qu. Rankexplorer: Visualization of ranking changes in large time series data. *IEEE TVCG*, 18(12):2669–2678, 2012.

[SDZ02] Ch Spathis, Michael Doumpos, and Constantin Zopounidis. Detecting falsified financial statements: a comparative study using multicriteria analysis and multivariate statistical techniques. *European Accounting Review*, 11(3):509–535, 2002.

[Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings, IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[sic13] Standard industrial classification (sic) system. http://www.census.gov/epcd/www/sic.html, 2013. Accessed: 2013-11-27.

[SP12] Standard and Poor's. Compustat data, 2012, (Accessed Feb 2012). http://www.compustat.com/compustat_data/.

[Spe04] Charles Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.

[SS04] Jinwook Seo and Ben Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *In Proceedings of the IEEE Symposium on Information Visualization*, pages 65–72, 2004.

[SSN+11] Hossam Sharara, Awalin Sopan, Galileo Namata, Lise Getoor, and Lisa Singh. G-pare: A visual analytic tool for comparative analysis of uncertain graphs. In *In Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 61–70, 2011.

[Suy09] S Suyanto. Fraudulent financial statement. *Gadjah Mada International Journal of Business*, 11(1):117–144, 2009.

[TAE+09] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnork, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 59 –66, 2009.

[TLKT09] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. Ensemblematrix: Interactive visualization to support machine learning with multiple classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1283–1292, 2009.

[TMF+12] Andrada Tatu, Fabian Maas, Ines Farber, Enrico Bertini, Tobias Schreck, Thomas Seidl, and Daniel Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology*, pages 63–72, 2012.

[Tuk77] John W. Tukey. *Exploratory data analysis*. Addison-Wesley, Reading, Massachusetts, 1977.

[Ulr13] Joshua Ulrich. *TTR: Technical Trading Rules*, 2013. R package version 0.22-0.

[Urs13] Anna Ursyn. *Perceptions of Knowledge Visualization: Explaining Concepts Through Meaningful Images*. IGI Global, Hershey, PA, USA, 1st edition, 2013.

[VBB12] Mohammad Valipour, Mohammad Ebrahim Banihabib, and Seyyed Mahmood Reza Behbahani. Parameters estimate of autoregressive moving average and autoregressive integrated moving average models and compare their ability for inflow forecasting. *J Math Stat*, 8(3):330–338, 2012.

[Vic11] Bret Victor. Up and down the ladder of abstraction. http://worrydream.com/LadderOfAbstraction/, 2011. [Online; accessed 01-June-2015].

[Wag82] Clifford H Wagner. Simpson's paradox in real life. *The American Statistician*, 36(1):46–48, 1982.

[WAG05] Leland Wilkinson, Anushka Anand, and Robert Grossman. Graph-theoretic scagnostics. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 157–164, 2005.

[War94] M.O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings of the IEEE Conference on Visualization*, pages 326 –333, 1994.

[War02] Matthew O Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002.

[WFYL08] Di Wu, Gabriel Pui Cheong Fung, Jeffrey Xu Yu, and Zheng Liu. Mining multiple time series co-movements. In *Proceedings of the 10th Asia-Pacific web conference on Progress in WWW research and development*, pages 572–583, 2008.

[WGG10] Yanhui Wu, Clive Gaunt, and Stephen Gray. A comparison of alternative bankruptcy prediction models. *Journal of Contemporary Accounting & Economics*, 6(1):34–45, 2010.

[WGK10] Matthew O Ward, Georges Grinstein, and Daniel Keim. *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010.

[WJ63] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

[WSJ+12] Pak Chung Wong, Han-Wei Shen, Christopher R Johnson, Chaomei Chen, and Robert B Ross. The top 10 challenges in extreme-scale visual analytics. *IEEE computer graphics and applications*, 32(4):63, 2012.

[XW+05] Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[YL03] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Learning*, page 856, 2003.

[YL04] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.

[YPH+04] J. Yang, Anilkumar Patro, S. Huang, Nishant Mehta, M.O. Ward, and E.A. Rundensteiner. Value and relation display for interactive exploration of high dimensional datasets. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 73 –80, 2004.

[YWR02] Jing Yang, Matthew O. Ward, and Elke A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 77–84, 2002.

[YWRH03] Jing Yang, Matthew O Ward, Elke A Rundensteiner, and Shiping Huang. Visual hierarchical dimension reduction for exploration of high dimensional datasets. In *Proceedings of the IEEE Symposium on Data Visualization*, pages 19–28, 2003.

[ZWRH12] Kaiyu Zhao, Matthew O Ward, Elke A Rundensteiner, and Huong N Higgins. Identifying descriptive data dimensions with integrated computational and visual methods. Unpublished manuscript, Nov 2012.

[ZWRH14] Kaiyu Zhao, Matthew O Ward, Elke A Rundensteiner, and Huong N Higgins. Lovis: Local pattern visualization for model refinement. In *Computer Graphics Forum*, volume 33, pages 331–340, 2014.

[ZWRH16] Kaiyu Zhao, Matthew O Ward, Elke A Rundensteiner, and Huong N Higgins. Mavis: Machine learning aided multi-model framework for time series visual analytics. *Visualization and Data Analysis*, 2016.