MaVis: Machine Learning Aided Multi-Model Framework for Time Series Visual Analytics

Kaiyu Zhao, Matthew Ward, Elke Rundensteiner, and Huong Higgins [*]

Worcester Polytechnic Institute

Abstract

The ultimate goal of any visual analytic task is to make sense of the data and gain insights. Unfortunately, the continuously growing scale of the data nowadays challenges the traditional data analytics in the ”big-data” era. Particularly, the human cognitive capabilities are constant whereas the data scale is not. Furthermore, most existing work focus on how to extract interesting information and present that to the user while not emphasizing on how to provide options to the analysts if the extracted information is not interesting. In this paper, we propose a visual analytic tool called MaVis that integrates multiple machine learning models with a plug-and-play style to describe the input data. It allows the analysts to choose the way they prefer to summarize the data. The MaVis framework provides multiple linked analytic spaces for interpretation at different levels. The low level data space handles data binning strategy while the high level model space handles model summarizations (i.e. clusters or trends). MaVis also supports model analytics that visualize the summarized patterns and compare and contrast them. This framework is shown to provide several novel methods of investigating co-movement patterns of timeseries dataset which is a common interest of medical sciences, finance, business and engineering alike. Lastly we demonstrate the usefulness of our framework via case study and user study using a stock price dataset.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces;

1 Introduction

Visual analytics nowadays has to deal with increasingly large scale data more often than ever in this ”big-data” era. One significant bottleneck for large-scale visual analytics is the human element within the analytic workflow [44]. While data scale is growing continuously and rapidly, the human cognitive abilities remain constant. The contradiction undoubtedly poses great challenges to the design of useful visual analytics systems that do not overload the analysts. To alleviate the cognition load, the data are often processed in a data reduction pipeline involving *binning*, *filtering*, *sampling*, *summarizing* and other variations [26]. Such data reduction process is a non-trivial task due to a chicken-and-egg dilemma. Namely, One, it has to capture the ”interestingness” of the data to provide an overview of the data space, while two, the ”interestingness” can often only be determined by the analysts after they ”see” the data. In a typical visual analytics process, the data reduction is often embedded in a user-driven *exploratory data analysis* [39] process where analysts experiment with different methods to gain insights by trail and error. However, this not only takes significant amount of time given the complexity and the growing scale of data nowadays, but also can be ineffective without approperiate visual support.

In order to address the above challenges, we propose a plug-and-play visualization framework that integrate multiple machine learning models to summarize the interestingness of the raw data. Four analytic spaces are provided to support this task and each of them is a specific scope for analytic tasks that are applicable to a particular type of objects such as 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. While in need high performance modern machine learning algorithms and a expectation of distributed infrastructures, dealing with large scale data is not our primary focus. The 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, *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?* To tackle the first issue, we design visual dinstinctions for the model descriptions that enables the analysts to swiftly determine what model to explore and. To deal with the second issue, we support the *exploratory data analysis* workflow of testing multiple methods and comparing them to reach a final conclusion. 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 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 [46, 32], 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. To get an overview of the data space, the MaVis *model relation space* also support the relationship analysis of local model descriptions.

However, investigating such phenomena can add complexity to the comparison analysis of the *model relation space* as there are, for instance, so 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. 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.

The main contributions of this work are as follows:

- **Explorations:** We design visualizations on top of machine learning tools to help reduce the scale of data. The plug-and-play models and multi-model comparison allows the user to explore the data from multiple angles with ease.

[*] e-mail: (kaiyuzhao,matt,rundenst,hhiggins)@wpi.edu

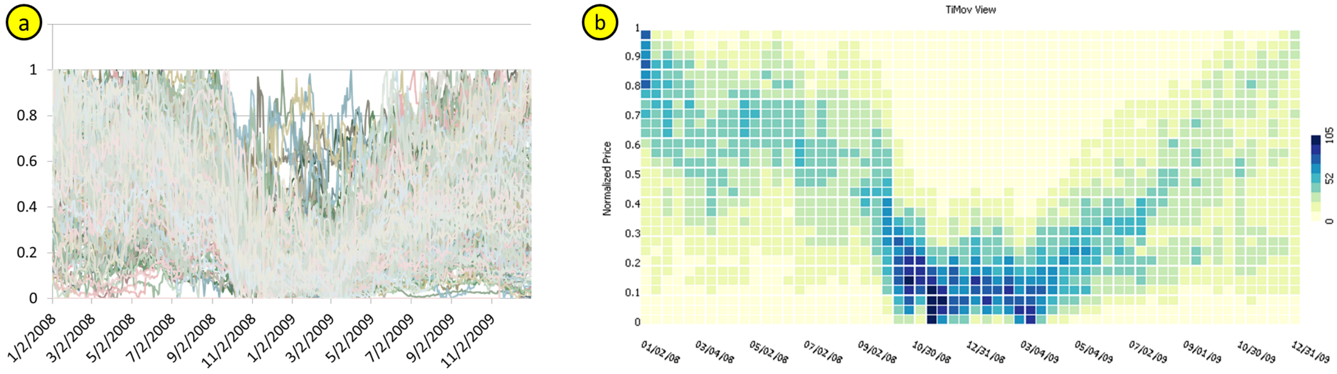

Figure 1: The line chart view (a) presents the data with the same normalization method (view rendered within Excel). *Time line movement* view (b) presents a collection of 250 time series where x-axis represents the time progression and y-axis is the normalized price values ranging from 0 to 1. The darker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

- **Analytics:** We provide 4 spaces, namely, data space, model space, relation space and nugget space to support analytics in MaVis. Each space supports specific analytic tasks such as data filtering and model comparisons. It also enables cross-space exploration so that analysts can link the findings in one space to another to gain more insights.
- **Evaluations:** We verifed that our MaVis framework can provide useful insights for co-movement analysis using stock price data in our case study. We also compare the effectiveness of alternative view design choices by analyzing the user performance and feedback after conducting a user study.

We discuss relevant machine learning techniques in Sec 2 followed by the framework design in Sec 3 discussing the cases of our 4 spaces. We provide evaluation in Sec 4, related work in Sec 5 and conclude the work in Sec 6.

2 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 model 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 [22] and detrended cross-correlation [35]. A derived space is then formed based on the extracted features such as frequency [15], trend [6], seasonality [11], and uncertainty [10] of the time series. The co-movement is a widely studied pattern of time series. The study of EEG co-movement in neuroscience [15] 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 [22]. The analysis of co-movement patterns in engineering can be used to optimize wireless device localization [12]. 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 [45] 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 [28], 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.

Next, we discuss three common types of models for time series data. Each of them are extracted by automated modeling techniques which are developed by other researchers.

2.1 Drift Model

Drift model is often used to describe the increasing or decreasing tendency of a non-stationary time series. It models the growth or decay of time series data, and in finance it is often used as an indication of whether longing or shorting a stock is likely making profit or not. Unlike the linear trend that describe the tendency as a function of time, a drift model usually describes the tendency change as a function of drift. Geometric Brownian motion [33] 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 [17]) may be used to estimate the parameters in the SDE, including the drift parameter θ. In our work, we integrate the pseudo-likelihood method implemented in R [17] into our system to extract the drift from time series data.

2.2 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 [23], economy, medicine [31] 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

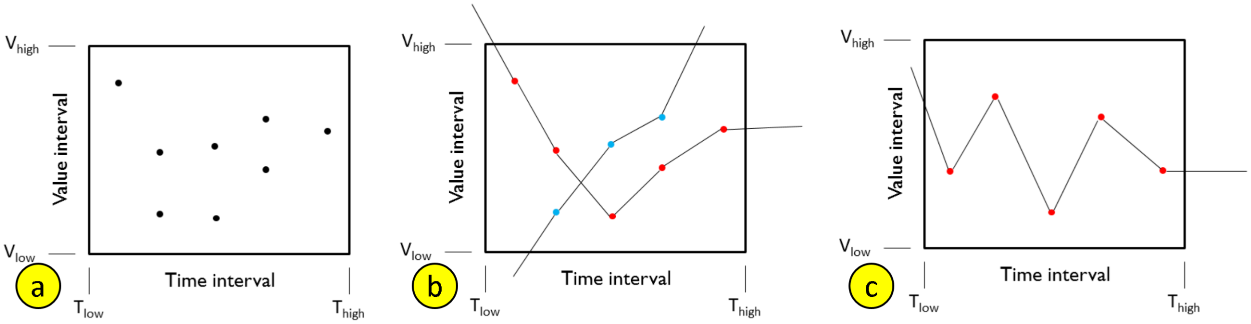

Figure 2: Comparison of two binning strategies for collection of time series. The binning method may count every data point (a) or count number of time series (b). Counting every data point of only one time series may lead to overemphasis of one bin due to the local fluctuations (c).

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 [29], ARIMA [42] and HP Filtering [21]. Since we focus on financial applications, we choose to integrate the ARIMA model parameter estimation [30] into our system. The ARIMA model can be used to estimate the most likely cycle duration of the time series and thus we use it here to represent the degree of co-movement regarding the seasonality duration.

2.3 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 [7]. It may also refers to the unpredictability of model parameters [5]. 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 [8].

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 the 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 [3] and implied volatility [1]. 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 [40] into our system.

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

3 MaVis Framework

In this section, we describe the design and implementation of the system that supports visual explorations in four spaces at different levels, 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* [41, 43] and the idea of *multi-scale representations* [25]. 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 after they learn the data items.

Any given models would however not always be perfect in terms of conveying accurate and useful insights. It is often unclear how well the model describes the original data [13] 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 can not be described by the global pattern [46]. We use a multi-scale representation strategy to model data at multiple granularities so that local pattern of interest is no longer lost. In order to support multiple granularities, MaVis provides user controled scales for capturing local patterns. The local patterns are then presented in a small multiple view to the analysts. Then, the local patterns and the global patterns may be compared and contrasted via the designed *linking operator*. Next we discuss in detail the design and implementation of the 4 spaces.

3.1 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 1a) (similar approach can be seen in [19]). Its variations such as ThemeRiver based design [36] 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 [26]. The binning strategy provies 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 at the beginning [19]. To overcome the clutter of the line chart view we design a *time line movement view* (as shown in Fig 1a). 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.

The *time line movement view* as presented in Figure 2 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 for higher density and ligher for lower. 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 [37]. 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*

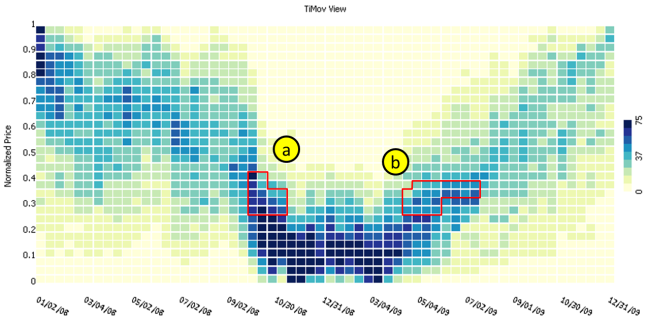

Figure 3: Two constraint boxes are placed to reveal companies that fell (a) and rise (b) during the 2008 crisis. Comparing to the view in Fig 1b, we see that most (70/ 100) of the prices move with such behavior. The color schema range is adjusted based on the max count of all the grid cells by default.

movement view is to count the number of values that fall into each grid cell (Figure 2a). 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 and the memory requirement is only 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 2b) that goes 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 2c). It requires extra memory to store the index of the time lines so that we only count for all duplications of each time line that bypasses a particular grid cell once.

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 [19] at the resolution level specified by the user 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, she may attach a small *negation* rectangle to the larger box(as shown in Fig 3).

The l*inking operator* links the user selection in the data space to model descriptions in the model space to further examine the co-movement of the selected time series regarding other domain specific features such as drift (for stock price analysis).

3.2 Model Space

In this section we focus on the three models we discussed in (Sec 2) 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 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 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.

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 [24], 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) 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 1b). Then the analyst may 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 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 interprete when line segments is overlapped in several regions. It is especially difficult to interprete when the density of line segments is high.

To overcome the above clutter issue we use a histogram view (Fig 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 4b), the color of all bars is updated accordingly to show the prevlance of the selection in other bins. It represents how these time series are distributed over the 4 local views (e.g., first half of 2006). The design for model space visualization are evaluated in our user study described in Section 4.2.

There are two types of brushing and linking operators in the model space. The first type is the linkage between multiple model

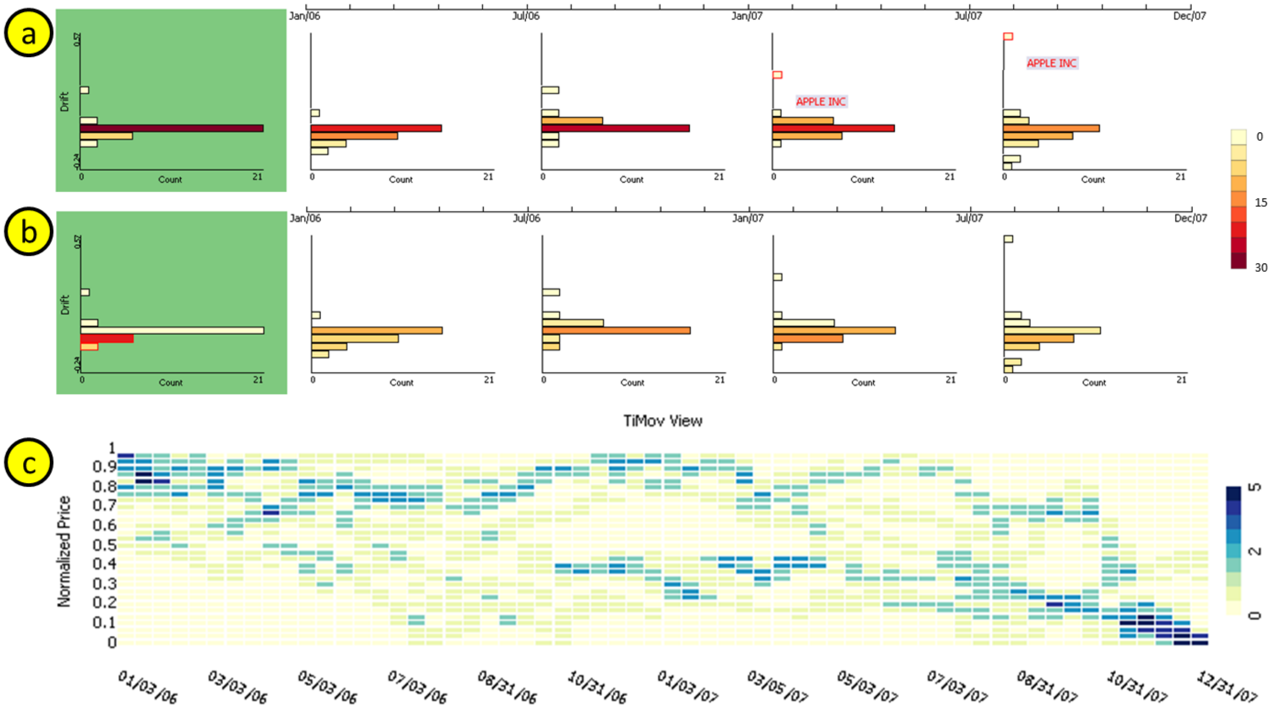

Figure 4: Drift abstraction of a collection of 32 time series objects. a) The default color encoding which represent the count of time series in each bin. b) Filter operator selects time series lower than the risk neutral zone. The color encoding represents the count of selected time series. c) Link the selected time series in b) back to 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 each view. In these set of views, we observe several interesting patterns. 1) most companies stays in the risk neutral zone which is the longest bar in all the histograms while many companies fell down towards the end of 2007. 2) We can also observe an outlier time series (Apple) that grows exceptionally well. 3) Linking the from 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.

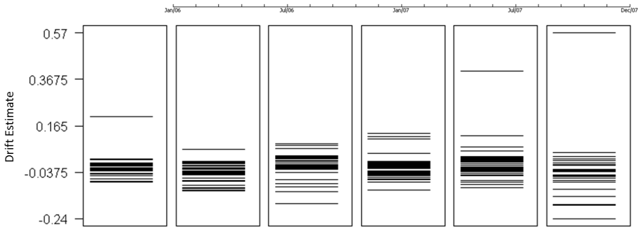

Figure 5: Time series similarity in the drift model space. The leftmost *bar code view* visualizes the overall drift tendency of the selected time series where each line corresponds to one time line. The 5 views to its right visualize the local drift.

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. *What are the volatilities of a selection of growing time series? How does the drift of a collection of time series change over time?* We discuss the design for analyzing the model relationships in detail in Section 3.3. The second type is the linking between model and 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 low drift estimate in the drift model space (Fig 4)b, the overal time line movement pattern is shown in the data space (Fig 4c).

3.3 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 contains every time series of the selection. We choose to use *Jaccard Similarity* as it is a commonly used measure for set similarities:

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

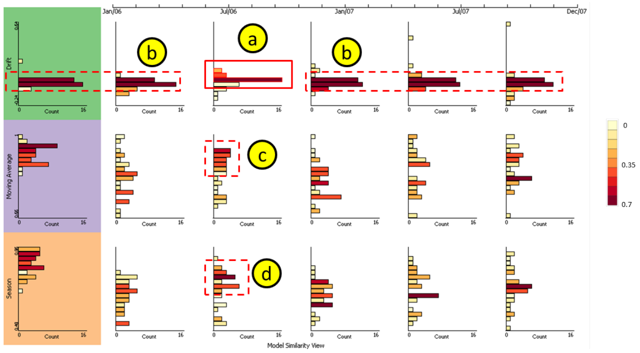

Figure 6: Model similarity analysis where a) is the brushed co-moving drift pattern starting in about July 2006. The darker color bins in b) shows high correlation between the brushed bins and other bins of different time intervals. The drift estimate of bins in a) and that in b) are at relatively the same value range. It shows the drift of co-moving patterns is quite consistent over time. Additionally, the darker color in c) indicates the selected group of time series have a high degree of volatility and a longer seasonal cycle d).

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

After computing the similarity, we update the color of bins (Fig 6) to represent it. In case of multiple bins are selected (e.g. 3 bins of time series are selected in Fig 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.

3.4 Nugget Space

The design of 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 differ over time or different model types. A pattern is defined by a user query and the difference is measured by the similarity between user queries. The objective of this analytic space is to answer these two questions and alike. *1) How closely are the current high risk (i.e., high volatility) relates to an increasing trend (i.e., high drift) in the future? 2) How many time series are present in such 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 achieve the above two goals by visualizing the summay information in a *nugget analytic view* (Fig 7) where the queries are compared and analyzed.

Nugget summarization: First, we discuss how to summarize and visualize a nugget that is created by a user query. For each nugget we need to present 3 types of information, the *time interval* of the user query, time series *distribution* for each model type, and the *model type* within which the analysts submit the query. 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. Inspired by the clockmap view [14], we use a round shaped glyph to present the summarization information (Fig 7). The outer space of the glyph is reserved to display the *time interval* of the user query. The inner space of the glyph displays the distribution of model descriptions of each of the three model types. 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

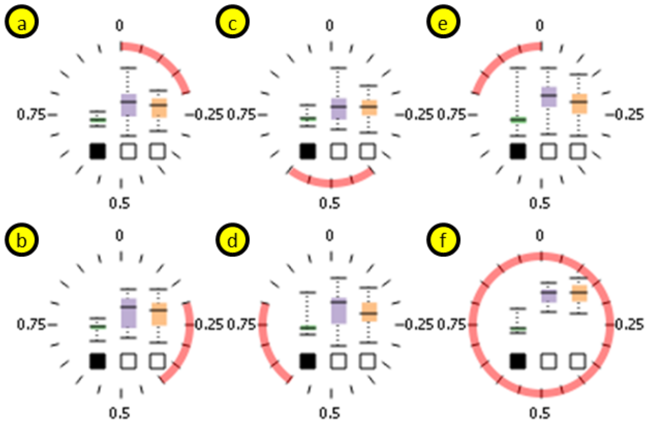

Figure 7: The view represents a collection of time series with co-moving trend that is identified in the first time interval indicated by the green box plot(a). However, the co-movement pattern of the same group became gradually diverging over the time and reaches peak during the last time interval (e). From long term aspect, the co-movement pattern is more consistent across three model types (f) comparing to the local diversities (a-e).

the user query that may lead to insights about the data. For example, in Fig 7c, even 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. It suggests that determining co-movement of the selected time series only by the drift is biased.

To visualize the summarization, we 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 4). We also experiment with the visual designs for indicating model types. We first use solid box-plot to indicate the user selected model type. In some cases, a user may be confused by the this method as the first quartile and third quartile may be very close to each other. In those cases, there is no way to highlight those boxes. Alternatively, we use a tickbox alike approach to make the view more consistent.

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. Let $\mathbf{v}_a$ and $\mathbf{v}_b$ be the vector representation of two patterns *A* and *B*. The similarity score is computed as:

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

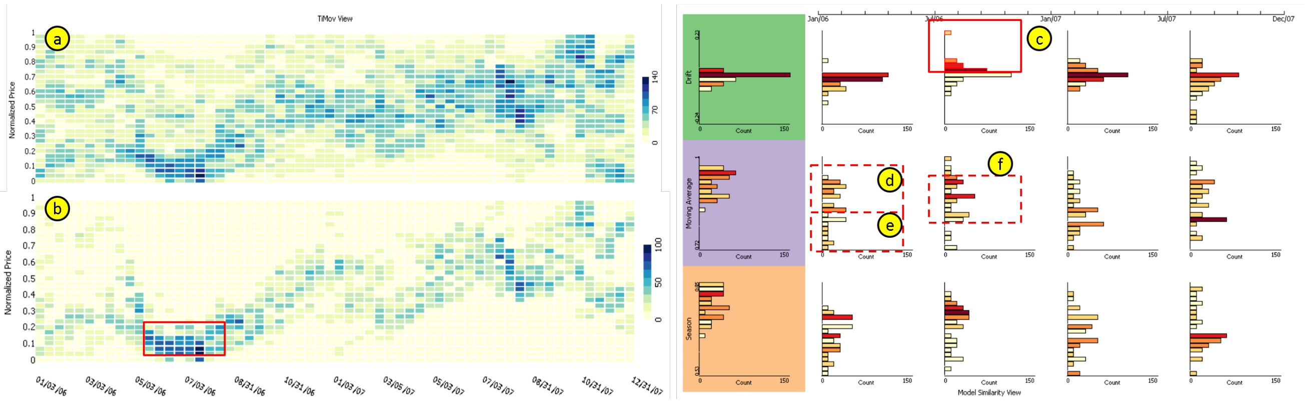

Figure 8: The views show a interactive exploration process for co-movement pattern investigation. The overall drift pattern is presented in a) and filtered results are shown in b) after a range query is submitted. In the view to the right, co-moving patterns are linked via color encoding. When the collection of growing time series are selected in c) the corresponding risk of this collection are linked to d) e) and f) where darker color in d) shows higher correlation and lighter color in e) shows lower correlation. The pattern in f) is also showing some degree of correlation but at high dispersion which means the collection is less likely co-moving.

The similarity measure above is a combination of pattern overlap measure (*Jaccard similarity coefficient*) and pattern summarization measure (*Euclidean distance*) while 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 on top of the n glyphs already displayed. 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 9 second row). The similarity score $s(a,b)$ is also displayed on the top left corner of each glyph.

4 System Evaluation

In this section, we discuss the evluation of MaVis framework using 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.

4.1 Case Study: Stock Price Co-movement

The purpose of the case study is to show that MaVis is able to support the discovering of patterns that are interesting to analysts, specifically people who often analyzing 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. The time series with missing values are discarded. After this cleaning process, out final collection conatins 348 companies and a total of 348,696 data points.

An analyst may have various questions she wishes to ask before starting the analysis of her data. 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 8a) to explore the data space. From the view, she perceives a dominant price fall pattern around *Jan. 2006 - June 2006*. She has a second question. *Does the selection of companies co-move in the other months?* She then submits a constraint query to preserve only the time series presenting a falling pattern before and near June 2006 (Fig 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. The selected collection of time series have an overall increasing trend in the data space according to the visual display.

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 8 right) to study model descriptions for the selected collection of time series. In Fig 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 8d & f) are correlated with the drift patterns. Specifically, the color encoding suggests that 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 get an overall sense of where to look next. To further analyze the dataset, she moves on to the *nugget analytic view* (Fig 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 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 the 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.

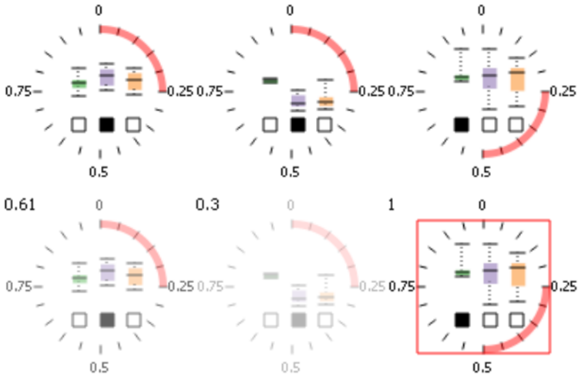

Figure 9: The first row (from left to right) shows the summary statistics of the selections in Fig 8d,e,c. The second row shows the same glyphs with focus in the item on the last column. The similarity score is calculated between the focused glyph and the other two glyphs and then applied to the alpha channel of all the glyphs.

4.2 User Study Design

We recruited 21 subjects including professors and students from the departments of Mathematics, Computer Sciences, and School of Business. The main purpose of this user study is to validate the *usefulness* and *design* of MaVis framework. 1) The usefulness test shows if MaVis is useful to an analyst for 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 comparing 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 how much 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 3 view designs of MaVis (3 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 an 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 fowllowing 3 questions:

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

1B Which of the following question may you want 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 the analysts to understand whether the comovement pattern in 2007 is consistent over time. 2) What are the name 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? Answer this question may help the analysts to get a broader picture about these companies such as understanding the volatililties and seasonal patterns. 4) Don't know. 5) Other.

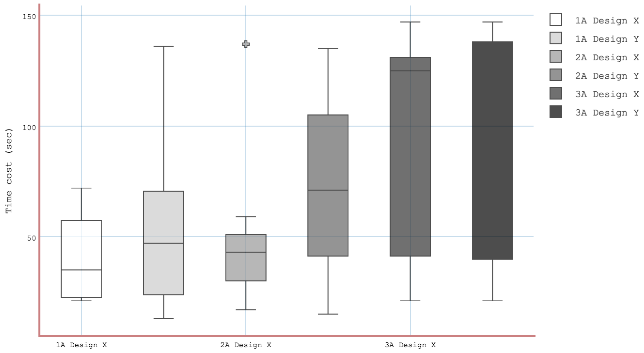

Figure 10: The chosen design of the views in question 1A and question 2A requires less time for discovering the pattern of interest. The two glyph views tested in question 3A require relatively the same amount of time. Hoever, the chosen design has better accurancy which is discussed in Sec 4.3.

1C Which design do you prefer in question *1A*, X or Y?

Typically, the choices for any questions are listed here. 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 gets no more questions and they don't know any other questions 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.

For question *1C*, we want to verify our design choices by learning the preference of each subject. For example, in question *1A* design X and Y are used. Specifically, based on the literature [2] for multivariate time series visualization techniques, line chart is the most approperate one to compare with our binned design. As it appears to have the highest information density compared to the other techniques such as ThemeRiver, Braided Graph and Circle view. The preference is discussed with the time and accurancy measure.

The questions for the other two views are in a similar style. We discuss the result in Sec 4.3. The other 6 quesions are designed to evaluate the *model similarity* and the *nugget analytic* view. The two design choices for the *model similarity view* are discussed in Sec 3.2 (barcode view vs. histogram). The two choices for the *nugget analytic view* are discussed in Sec 3.4 (linear space vs. circular space).

4.3 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 each subject for both design X and Y are summarized in Fig 10. It shows the time

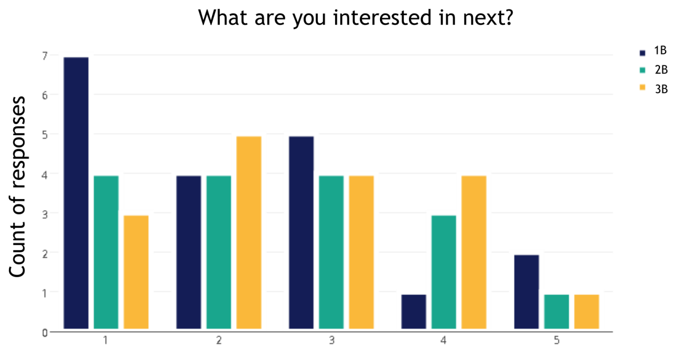

Figure 11: Each question B has 5 options (x axis) a subject may choose from (Sec 4.2). Option 1 to 3 for question B are supported by our system and the user may dig further to discover more insights. Response 4 is *Don't know* which means the subject may have no more questions. Option 5 is *Other* and the user may have additional questions to query the system but we do not support yet. Based on the result, few subjects chose option 5 indicating the framework covers most their futher needs initiated from the given 3 questions.

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 *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: [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.

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 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 and these are in-depth questions such as ”why do all the companies drop at the same time?”. To answer these questions, the analysts may need more analysis and it is beyond the scope of our toolkit. Most of the subjects selected questions that can be answered by the system. It shows that our system works as expected and it is able to guide the user to further investigate pattern of interest during the exploration process. More subjects tend to choose option 4 in higher analytic spaces. As we can see in Fig 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% which confirms that we made reasonable choice for our final design.

5 Related Work

Recently, several work have attempted to utilize model-driven visualization to help analyzing data. The model-driven approach by Garg et. al. [16] 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 that it aims to reduce visual complexity using algorithmic methods. MaVis provides multiple automated modeling methods for reduction and additionally allow comparison and contrast between them to gain more insights. Dis-Function [9] presents a system to learn the distance between data objects with both user input and predefined metrics. It handles the low-level optimization such as distance computing and presents high-level patterns to the user. 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 way of measuring similarity. The Nugget Browser [18] displayes 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 support 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 [38] 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 are instead designed for data exploration while using modeling techniques for data reduction. Potter et. al. [34] 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. The authors of CVVs [20] explored visual design spaces for presenting correlated visual representations in case of complex and 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 are also relavent to our work. Show Me [27] 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. [4] 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.

6 Conclusion and Future Work

In this paper, we present the MaVis framework. It is a system designed for identifying co-movement patterns from time series dataset. It provides 4 analytic spaces that allow the analyst to navigate between them. It integrates multiple models to support the interpretation of 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 and conducted user study. There are several interesting future directions based on this work. First, the models for data reduction can potentially be extended to support stream data summary. Second, this framework can be potentially extended to support time series forcasting. Third, the data modeling process can be integrated with visual interactions so that the automatic data reduction can be guided and adjusted by human experts.

References

[1] W. Abdelmalek, S. Ben Hamida, and F. Abid. Selecting the best forecasting-implied volatility model using genetic programming. *Advances in Decision Sciences*, 2009, 2009.

[2] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, London, 2011.

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

[4] 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.

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

[6] B. A. Blonigen, J. Piger, and N. Sly. Comovement in gdp trends and cycles among trading partners. *Journal of International Economics*, 94(2):239–247, 2014.

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

[8] K. Brodlie, R. A. Osorio, and A. Lopes. A review of uncertainty in data visualization. In *Expanding the Frontiers of Visual Analytics and Visualization*, pages 81–109. 2012.

[9] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Visual Analytics Science and Technology (VAST)*, pages 83–92, 2012.

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

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

[12] G. Chandrasekaran, M. A. Ergin, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen. Decode: Exploiting shadow fading to detect comoving wireless devices. *Mobile Computing, IEEE Transactions on*, 8(12):1663–1675, 2009.

[13] Q. Cui, M. O. Ward, E. A. Rundensteiner, and J. Yang. Measuring data abstraction quality in multiresolution visualizations. *IEEE TVCG*, 12(5):709–716, 2006.

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

[15] C. Flamm, A. Graef, S. Pirker, C. Baumgartner, and M. 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.

[16] S. Garg, J. E. Nam, I. Ramakrishnan, and K. Mueller. Model-driven visual analytics. In *Visual Analytics Science and Technology (VAST)*, pages 19–26, 2008.

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

[18] Z. Guo, M. O. Ward, and E. 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.

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

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

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

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

[23] M. J. Kamstra, L. A. Kramer, and M. D. Levi. A careful re-examination of seasonality in international stock markets: Comment on sentiment and stock returns. *Journal of Banking & Finance*, 36(4):934–956, 2012.

[24] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In *Data mining in Time Series Databases. Published by World Scientific*, pages 1–22, 1993.

[25] R. 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.

[26] Z. Liu, B. Jiang, and J. Heer. immens: Real-time visual querying of big data. In *Computer Graphics Forum*, volume 32, pages 421–430, 2013.

[27] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE TVCG*, 13(6):1137–1144, 2007.

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

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

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

[31] R. Moineddin, R. Upshur, E. Crighton, and M. Mamdani. Autoregression as a means of assessing the strength of seasonality in a time series. *Popul Health Metr*, 1(1):10, 2003.

[32] T. Muhlbacher and H. Piringer. A partition-based framework for building and validating regression models. *IEEE TVCG*, 19(12):1962–1971, 2013.

[33] M. Pinsky and S. Karlin. *An introduction to stochastic modeling*. Academic press, Oxford, UK, 2010.

[34] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *Data Mining Workshops, ICDMW*, pages 233–240, 2009.

[35] J. C. Reboredo, M. A. Rivera-Castro, and G. F. Zebende. Oil and us dollar exchange rate dependence: A detrended cross-correlation approach. *Energy Economics*, 42:132–139, 2014.

[36] C. Shi, W. Cui, S. Liu, P. Xu, W. Chen, and H. Qu. Rankexplorer: Visualization of ranking changes in large time series data. *IEEE TVCG*, 18(12):2669–2678, 2012.

[37] B. 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.

[38] J. Talbot, B. Lee, A. Kapoor, and D. 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.

[39] J. W. Tukey. Exploratory data analysis. 1977.

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

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

[42] M. Valipour, M. E. Banihabib, and S. M. R. 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.

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

[44] P. C. Wong, H.-W. Shen, C. R. Johnson, C. Chen, and R. B. Ross. The top 10 challenges in extreme-scale visual analytics. *IEEE computer graphics and applications*, 32(4):63, 2012.

[45] D. Wu, G. P. C. Fung, J. X. Yu, and Z. 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.

[46] K. Zhao, M. O. Ward, E. A. Rundensteiner, and H. N. Higgins. Lovis: Local pattern visualization for model refinement. In *Computer Graphics Forum*, volume 33, pages 331–340, 2014.