STAC: Enhancing Stacked Graphs for Time Series Analysis

Yun Wang * Tongshuang Wu† Zhutian Chen‡ Qiong Luo§ Huamin Qu¶

The Hong Kong University of Science and Technology

Abstract

Stacked graphs have been widely used to represent multiple time series simultaneously to show the changes of individual values and their aggregation over time. However, when the number of time series becomes very large, the layers representing time series with small values take up only very small proportions in the stacked graph, making them hard to trace. As a result, it is challenging for analysts to detect the correlation of individual layers and their aggregation, and find trend similarities and differences between layers solely with stacked graphs. In this paper, we study the correlations of individual layers, and their aggregation in time series data presented with stacked graphs, focusing on the local regions within any given time intervals. Specifically, we present STAC, an interactive visual analytics system, to help analysts gain insights into the correlations in stacked graphs. While preserving the original stacked shape, we further link a stacked graph with auxiliary views to facilitate the in-depth analysis of correlations in time series data. A case study based on a real-world dataset demonstrates the effectiveness of our system in gaining insights into time series data analysis and facilitating various analytical tasks.

1 Introduction

Stacked graphs have been widely adopted for visualizing time-varying data in various fields, such as email messages analysis [11], tracking news stories [12], music listening histories analysis [4] and so on. Compared with more widely used line graphs, which show the fluctuation of each time series, stacked graphs are more useful for concurrently showing the proportion of each time series in the aggregation, as well as the volume changes. By stacking one layer onto another, stacked graphs can be used to avoid the overlap of time series in line graphs and thus have better scalability with number of time series [14].

While stacked graphs can visualize hundreds of time series concurrently [29], analysts might still find it difficult to conduct correlated analysis among layers. First, since the aggregation distorts the baselines of the layers, identifying the trend of each individual layer is non-trivial, and it is even harder to spot the layer-to-layer relationship in a stacked graph. For example, from the stacked graph in Fig. 1(a), we cannot easily pinpoint the fluctuations and the similarities of A and B (as well as C and D). Second, the relationships between individual layers and their aggregation remain unclear. This problem leads to ambiguity in understanding how the aggregated shapes are formed: it is hard to know if an aggregated shape is constructed by a large number of layers with similar trends, or only represents the shape of one or two outliers. For example, in Fig. 1, the dent at time t is mainly caused by layer A and B, whereas the other series have much less contribution to the aggregation. Layers C and D, which contain bumps at this point, are not obvious in Fig. 1(a). Third, the complexity of stacked graphs limits the recognition of local time-patterns. Since stacked graphs are mainly used for time series data, it is very likely for layers to span long in time. While the overall trends might be easy to spot, some small yet potentially valuable variations could easily be overlooked.

*e-mail: ywangch@cse.ust.hk

†e-mail:twuac@stu.ust.hk

‡e-mail:zchenbn@cse.ust.hk

§e-mail:luo@cse.ust.hk

¶e-mail:huamin@cse.ust.hk

Figure 1: (a) An aggregated dent at t. (b) The involved layers A, B, C and D contribute differently. A and B are similar to the aggregation showing a dent while C and D have an opposite trend showing a bump.

To tackle these problems, we introduce *STAC* in this paper, a visual analytics system that augments stacked graphs with additional visual representations and intuitive interactions, so that users could gain more insight into the trends of individual time series, the relations between layers and their aggregation, which are not obvious in traditional stacked graphs. Specifically, STAC reduces ambiguities in stacked graphs ("S" in STAC) with three additional linked views, namely "Trend view", "Aggregation view" and "Correlation view". Moreover, with a sliding time window, analysts can interactively explore how multiple time series change during a certain time period and how they vary over a large time scale.

The main contributions of this paper are summarized as follows:

- Examine the problems for analyzing multiple time series with stacked graphs, especially the relations among multiple time series, the aggregation, and their changes over time.
- Implement an interactive system with multiple novel views to show the flows of stacked graphs, which can reveal relations of layers in different time scales.
- Conduct a case study based on a real-world dataset, which demonstrates the usefulness of our system, and discuss the insights that could be gained by using our system.

2 Related Work

In this section, we provide an overview of related research for time series data mining and time series data visualization.

2.1 Time Series Data Mining

A time series records a set of values collected over time. As such, the size of time series data grows with time [22]. Typical examples include economic forecast [21], intrusion detection [6], gene expression analysis [8], medical surveillance [23] and hydrology [31]. Among all the techniques used in time series data mining, clustering [18] is perhaps the most frequently used. However, when the time span of a time series is long, it is impractical to use whole time series clustering for data analysis, and sub-sequence time series (STS) clustering is a viable alternative [20]. In our work, we apply STS clustering to support multiple time segments analysis and adapt DBSCAN [13] to cluster the time series.

2.2 Time Series Visualization

Visualization of time series can be done through various methods [1, 2]. For example, line graphs are most used to show multiple time series [28]. With time and values mapped to a common baseline, it is easy to compare the differences of values and trends. However, line graphs are only capable of comparing multiple time series when the number of time series is quite small [14]. To avoid overplotting, small multiples are introduced to show several time series [28], where each time series has its own baseline. With small multiples, users can have a clear overview of the changes of several time series over time. However, it is not space efficient and comparisons between time series remain difficult.

Stacked graphs are used to simultaneously show each layer and their aggregation with one common timeline [29]. With stacked graphs, analysts can clearly analyze a large number of time series and their aggregation, and detect the time series with larger data values. Many studies have focused on improving stacked graphs from different aspects and have applied stacked graphs to various fields. ThemeRiver is proposed to depict temporal thematic change with smooth curves in a symmetrical style [12]. Later on, Byron and Wattenberg [4] introduced Streamgraphs to improve the legibility and aesthetics of stacked graphs. In addition to aesthetics and legibility, utilities of stacked graphs have long been improved by combining with additional tools and visual elements. For example, NameVoyager [29] improves the exploration of stacked graphs by combining search and filtering with stacked graphs. Subsequently, TouchWave [3] is presented to enable layout adjustments by implementing multi-touch capabilities on multi-touch devices. TIARA [30], Cloudlines [16], TextFlow [5], and EvoRiver [26] are exploratory text analytics systems that integrate text mining techniques to explore large collections of text data and to reveal the evolution of topics over time. Apart from text data, stacked graphs are also used for ranking changes in search engine data and other data by embedding color bars and transition glyphs to augment time series data analysis [24, 25]. Other applications include personalized recommendations of stacked graphs [27] and interactive stacked graph systems, enabling updating information stream [7]. Although previous systems have used rich visual encoding from different aspects, none of them can support efficient exploration of correlations among multiple time series, which is essential for our time series data exploration. Some recent work also aims to facilitate the analysis of correlations in time series data. However, they either focus on time series curves instead of stacked graphs [15] or adopt an integrated design that does not support comprehensive correlation analysis with multiple time series and time segments [32]. In this paper, we address a similar problem with quite a different approach. Our system integrates several well-designed visualization techniques into a linked view system to help people understand the construction of stacked graphs while maintaining the advantages of using stacked graphs to present time series data.

3 Requirement Analysis

In previous work, stacked graphs have been extensively used, as users can easily locate the layers with the most contribution to the trend of the aggregation. However, the volumes and trends of less significant time series should not be ignored.

3.1 Analytical Tasks

We list four types of analytical tasks and further compile a set of design rationales.

T.1: Individual Time Series: A basic requirement of analyzing time series data is to show how each time series changes over time. Analysts should be able to understand how the data values of individual time series changes over time. In addition to the values of each time series, analysts should know the trend of individual time series to understand how they develop.

T.2: Time Series Aggregation: It is also important to study the relations between individual time series and the aggregation of a set of time series. When studying time series aggregation, analysts need to know the individual contributions of an aggregation. More specifically, the time series with positive and negative contributions to the whole aggregation should be identified.

T.3: Multiple Time Series: Correlation exploration of multiple time series is crucial for time series data analysis. Analysts need to understand the relations of individual time series in multiple time scales. More specifically, the similarity of any two time series should be visualized. In addition, the clusters and the changes of clusters of multiple time series should be extracted to help analysts understand the evolution of time series over a long period of time.

T.4: Time Interval segmentation: As the relations of time series can change greatly over a large time scale, time interval segmentation should be properly done to help analysis of the changes over time. With time interval segmentation, the exploration of partial time scale can be done at different time scales. Furthermore, the analysis using different time intervals should be facilitated.

3.2 Design Rationale

Based on the identified analytical tasks, we further compile a set of design principles of developing our time series analytics system:

R.1: Multiple Time Series Exploration: As discussed, the analysis of time series data can be divided into four components, i.e., individual time series, multiple time series, time series aggregation and time interval segmentation. The system should allow users to easily recognize patterns from different perspectives. The system should also support multiple well-coordinated views to enable joint analysis.

R.2: Multiple Time Segments Analysis: The multi-scale analysis of the time series includes two components: support of multiple time points/scales and partial exploration of time series datasets. To analyze and identify the patterns between scales, the system should support visual comparisons between different time windows and subsets of time series data.

R.3: Interactive Visual Feedback: The analysis of time series requires a trial-and-error process. It is crucial for analysts to interact with data directly. In addition, analysts may have different requirements and need further exploration of certain properties of data. It is important to support interactions that can filter data in the process of analysis.

R.4: Intuitive Visual Narrative: By supporting intuitive visual narrative, it is easier for analysts to follow a predefined internal logic of the analysis and presentation of visual patterns. Analysts may have requirements for the customization of the internal logic of the analysis of different perspectives. With an intuitive narrative style, analysts can more effectively present their findings and communicate with each other to iteratively improve their work.

4 Visualization Design

Motivated by the identified requirements, we have designed four sub-views to support correlated analysis.

4.1 Sub-view Design

Stacked Graph View: The Stacked Graphs View is the essential view for users to have a basic idea of the original data. With this view, the advantages of concurrently viewing both individual time series and aggregations can be maintained, and thus, users could easily grasp an overview of the whole time series.

Trend View: It is hard to understand relative variations of individual time series when their data values are small. To make their trends easier to assess, we calculate the slopes of each individual time series at every time point, and take the output as the trend. Specifically, we normalize a time series by subtracting the mean

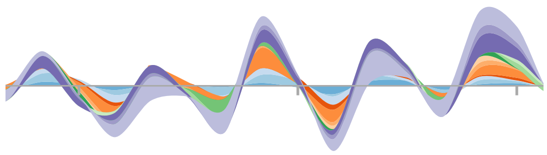

Figure 2: Trend view: each layer encodes the rate of value change for each time series.

value in the sequence and dividing all values by the standard deviation to extract the trends. This normalization enhances the shape aspect of the sequence [9]: $y = (y - \bar{y})/\sigma$. It further enhances the local features by maximizing their curvatures, which is also commonly adopted in STS clustering [10]. We choose stacked graphs instead of commonly used line graphs for its scalability: line graphs overlay on each other and the trend will be hidden.

Aggregation View: Another important goal is to show the relations between an individual time series and the aggregation. Analysts need to know the contributors and internal structures of the aggregation. Because of the variation in long time sequences, the similarities of time series may change greatly at different time points. Thus, we use time segmentation to allow analysts to understand how each group changes at different time intervals.

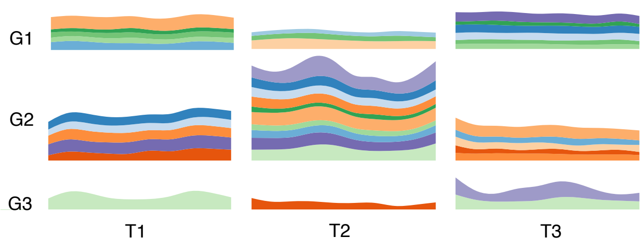

Figure 3: Aggregation view: in each time segment, time series are divided into groups based on their similarity to the aggregation.

By default, we divide the time series within each time segment into three groups based on the similarity between each individual layer and the aggregation. The range of similarities is divided into three equal subranges. The layers whose similarities fall into the three subranges are grouped accordingly. The first group of time series data has a consistent trend with the aggregation: they follow the same increasing or decreasing trend of the corresponding aggregation. This group of time series has contributions to the whole trend. In contrast, the time sequences in the second group would have unclear contributions to the aggregation, and the third group has an opposite trend to the aggregation. When the aggregation increases, the time series decreases, and vice versa. Users can also adjust the grouping criterion for different needs. After that, each group of layers is aggregated. Users can identify the volume and the number of layers of each group. The layers are colored as in the Stacked Graphs View, such that we could see how the aggregation of each group changes, as well as how the layers are classified into different time segments (Fig. 3).

Correlation View: To demonstrate the relations among multiple time series, we use a correlation view to show the relations of all the time series. We choose the widely used Dynamic Time Warping (DTW) to calculate the similarity of time series data [19]. After that, we use DBSCAN to generate clusters of time series data. Since the correlation of a large number of time series may change considerably over a long period of time, it becomes less useful to cluster multiple time series through a large time scale. Therefore, we segment the time series into smaller time intervals to capture local similarities within one interval, as well as between multiple intervals. To support an intuitive and effective exploration of relations among multiple time series, we use a multidimensional scaling (MDS) [17] layout, which is well-known for revealing the distributions of the time series data, to visualize the relations among time series. Then, we use grey outlines as visual cues to enhance the display of clusters.

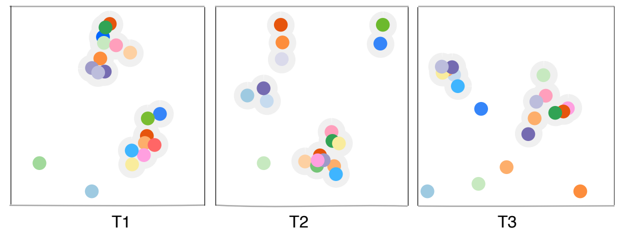

Figure 4: Correlation view: in each time segment, an MDS layout shows the distribution of the time series. Closer time series reflect more similarity.

4.2 Interaction

Based on the design rationale, our system should enable analysts to interact with data and facilitate exploration with flexibility. Specifically, STAC supports the following interactions:

Linking: The system supports linking among the four views to facilitate comprehensive visual feedback. It also facilitates multiple time series exploration. For example, if a time series in the aggregation view is selected, the corresponding time series in the stacked graph view, trend view, aggregation view and correlation view will be highlighted.

Filtering: Filtering enables analysts to focus on important information, and eliminates less important time series to help analysts handle large scale and uncertain datasets. STAC allows analysts to select important layers and eliminate less important ones through different views.

Customization: Analysts can configure our system by choosing parameters, such as the grouping criterion for the Aggregation View. Analysts can also segment a long time period by selecting a shorter time interval. To support intuitive visual narrative, the order of the four views can be rearranged by analysts to cater their own needs.

5 System Overview

We design our system to allow users to explore and analyze time series. The framework of our system consists of three modules: data preprocessing, data modeling, and visual analysis.

Data preprocessing extracts time series data from raw datasets, and stores time series data in a unified form. The data analysis module computes the trend of the time series, the similarity between two time series, and groups the time series based on their similarity to the aggregation using dynamic time warping [19]. In order to support multidimensional scaling (MDS) [17] in the Correlation view, we calculate the similarity between each time series pair with DBSCAN [13] to cluster the time series, because it is the most representative density-based clustering algorithm. In addition, we compute the similarity of each time series to the aggregation to support classification of all the time series into three types for an aggregation view. When users select a time window, we send the data to the back-end and recalculate the results for visualization to support interactive time segmentation.

The visual analysis module provides a visual interface integrating four linked views to help users explore multiple time series data. All the four views of our system are placed horizontally to fit into the wide screen of modern monitors better, with the timeline starting from left to right. The four views align vertically with the same time axis to enable interactive joint analysis (Fig. 6). With a control panel on the left, users can customize the datasets in use, the time range for analysis, and the viewing sequence of the four views. To support interactive joint analysis, we allow users to configure time

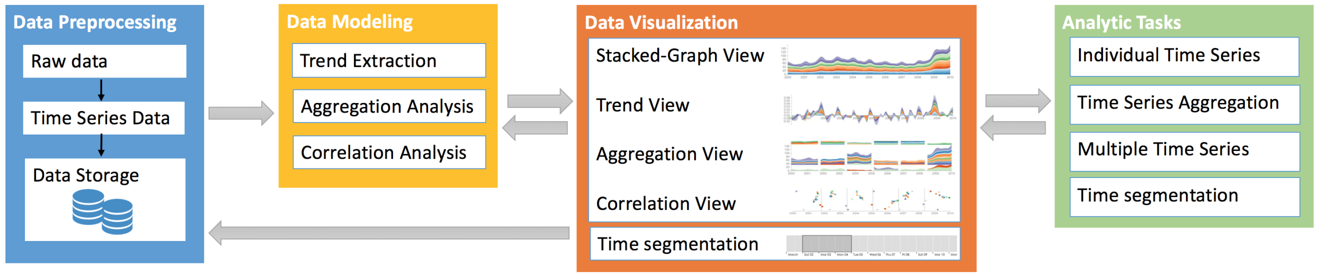

Figure 5: System workflow: In the data preprocessing phase, we extract time series from raw data. In the data modeling phase, we conduct trend extraction, aggregation analysis and correlation analysis. In the visual exploration phase, four coordinated views are provided to support four basic analytic tasks.

interval segmentation by selecting a certain range of time scale as a time window (t_i, t_j). Based on the selection, the time interval can be calculated ($T = t_j - t_i$), and the time axis is segmented into many time windows (Eq. (1)). Through time segmentation, users can easily compare and trace the differences of the patterns over time:

$$T_{axis} = \ldots \cup (t_i - T, t_i) \cup (t_i, t_j) \cup (t_j, t_j + T) \cup \ldots \quad (1)$$

6 Results

6.1 Implementation

We have conducted experiments to explore real-world time-series data and give a case study to show how the system works. Our system is implemented as a web-based application that allows users to interact with preprocessed data. We use MEAN.JS with MongoDB, Express, AngularJS, and Node.js, to build an interactive visualization system. We integrate the four views implemented in the front-end visualization with D3.js.

6.2 Case Study

We use the unemployment rates of 14 industries from 2000 to 2010 collected by the US Bureau of Labor Statistics to analyze if the rates among industries share similar variations, or if the summation is similar to individual rate sequences.

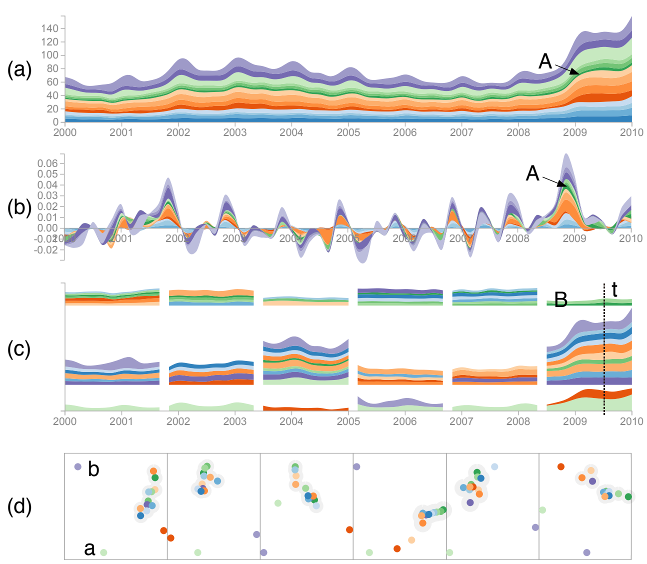

Figure 6: Visualization generated by STAC for the unemployment rates of 14 industries. The original stacked graph is shown in (a), followed by (b)trend view, (c)aggregation view and (d)correlation view.

We first notice that in the trend view (Fig. 6(b)), layer A (Government) reaches a significant height around 2009, whereas in Fig. 6(a), it is nearly unnoticeable. This phenomenon suggests that while the unemployment rate represented by this layer does not constitute a large proportion compared to the others, it varies greatly, revealing an unstable unemployment state, which could be ignored easily with stacked graphs alone.

The aggregation view is also informative. We configure the aggregation view by adjusting the three subranges of similarity and classifying the layers into “highly similar”, “somewhat similar” and “not similar”. After navigation, we first notice that there is at least one group in each period whose aggregated shape is flat. This finding indicates no matter how the aggregation varies, the unemployment rates of some industries are in fact reasonably stable. Another example for the differences between the aggregation and the layers is shown in Fig. 6(c). At point t, we could see a small bump which is not in the “highly similar” group. From the observation, we find it is the result of the layers in cluster B. This could be an indicator for users to adjust time segments to perform more detailed analysis.

We also notice a clustering phenomenon in the correlation view, as in Fig. 6(d). For instance, in interval T_1, almost all points representing layers are placed on the right except points a and b. Further looking around these two points, we find they are far from the other points in all the intervals. Referring back to the original data, we find that these two, representing Agriculture and Construction, respectively, indeed vary far more significantly than the other layers, showing unstable unemployment rates.

7 Conclusion

In this paper, we propose STAC, an interactive visual analytics system to facilitate analysts using stacked graphs to better understand time series data. We use four views to show how time series and their aggregation evolve with time. Results show that STAC can effectively reveal patterns and help analysts assess multiple time series from different aspects.

Although useful and effective, STAC has some design limitations. A major problem is that for the four views, we use the same color encoding for time series, which limits the number of time series that can be encoded. We will further extend STAC to improve its scalability. Moreover, to demonstrate the evolutions and the groupings of the layers over time, we plan to improve the visual analytical components by connecting layers between time segments.

Acknowledgements

The authors would like to thank Dr. Xiaojuan Ma, Dr. Weiwei Cui, Dongyu Liu, Yuanzhe Chen, Haipeng Zeng, Quan Li for their help for this project, and the anonymous reviewers for their valuable comments. This work is partially supported by the National Basic Research Program of China (973 Program) under Grant No. 2014CB340304.

References

[1] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60, 2008.

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

[3] D. Baur, B. Lee, and S. Carpendale. Touchwave: kinetic multi-touch manipulation for hierarchical stacked graphs. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, pages 255–264. ACM, 2012.

[4] L. Byron and M. Wattenberg. Stacked graphs–geometry & aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, 2008.

[5] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. J. Gao, H. Qu, and X. Tong. Textflow: Towards better understanding of evolving topics in text. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2412–2421, 2011.

[6] D. E. Denning. An intrusion-detection model. *IEEE Transactions on Software Engineering*, (2):222–232, 1987.

[7] M. Dörk, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1129–1138, 2010.

[8] J. Ernst and Z. Bar-Joseph. Stem: a tool for the analysis of short time series gene expression data. *BMC bioinformatics*, 7(1):191, 2006.

[9] S. J. Gaffney. *Probabilistic curve-aligned clustering and prediction with regression mixture models*. PhD thesis, University of California, Irvine, 2004.

[10] D. Goldin, R. Mardales, and G. Nagy. In search of meaning for time series subsequence clustering: matching algorithms based on a new distance measure. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 347–356. ACM, 2006.

[11] S. Hangal, M. S. Lam, and J. Heer. Muse: Reviving memories using email archives. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 75–84. ACM, 2011.

[12] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[13] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[14] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):927–934, 2010.

[15] P. Köthur, C. Witt, M. Sips, N. Marwan, S. Schinkel, and D. Dransch. Visual analytics for correlation-based comparison of time series ensembles. In *Computer Graphics Forum*, volume 34, pages 411–420. Wiley Online Library, 2015.

[16] M. Krstajić, E. Bertini, D. Keim, et al. Cloudlines: Compact display of event episodes in multiple time-series. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2432–2439, 2011.

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

[18] T. W. Liao. Clustering of time series data: a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

[19] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[20] T. Oates. Identifying distinctive subsequences in multivariate time series by clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 322–326. ACM, 1999.

[21] R. S. Pindyck and D. L. Rubinfeld. *Econometric models and economic forecasts*, volume 4. Irwin/McGraw-Hill Boston, 1998.

[22] C. A. Ratanamahatana, J. Lin, D. Gunopulos, E. Keogh, M. Vlachos, and G. Das. Mining time series data. In *Data Mining and Knowledge Discovery Handbook*, pages 1049–1077. Springer, 2010.

[23] B. Y. Reis and K. D. Mandl. Time series modeling for syndromic surveillance. *BMC Medical Informatics and Decision Making*, 3(1):2, 2003.

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

[25] C. Shi, Y. Wu, S. Liu, H. Zhou, and H. Qu. Loyaltracker: Visualizing loyalty dynamics in search engines. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1733–1742, 2014.

[26] G. Sun, Y. Wu, S. Liu, T.-Q. Peng, J. J. Zhu, and R. Liang. Evoriver: Visual analysis of topic coopetition on social media. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1753–1762, 2014.

[27] A. Toledo, K. Sookhanaphibarn, R. Thawonmas, and F. Rinaldo. Personalized recommendation in interactive visual analysis of stacked graphs. *ISRN Artificial Intelligence*, 2012, 2012.

[28] E. R. Tufte and P. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.

[29] M. Wattenberg. Baby names, visualization, and social data analysis. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 1–7. IEEE, 2005.

[30] F. Wei, S. Liu, Y. Song, S. Pan, M. X. Zhou, W. Qian, L. Shi, L. Tan, and Q. Zhang. Tiara: a visual exploratory text analytic system. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 153–162. ACM, 2010.

[31] A. W. Wood, L. R. Leung, V. Sridhar, and D. Lettenmaier. Hydrologic implications of dynamical and statistical approaches to downscaling climate model outputs. *Climatic change*, 62(1-3):189–216, 2004.

[32] T. Wu, Y. Wu, C. Shi, H. Qu, and W. Cui. Piecestack: Toward better understanding of stacked graphs' formation. *IEEE Transactions on Visualization and Computer Graphics*, 2016.