Mining, Pruning and Visualizing Frequent Patterns for Temporal Event Sequence Analysis

Zhicheng Liu, Himel Dev, Mira Dontcheva and Matthew Hoffman

Abstract—Integrating frequent pattern mining with interactive visualization for temporal event sequence analysis poses many interesting research questions and challenges. We review and reflect on some of these challenges based on our experiences working on event sequence data from two domains: web analytics and application logs. These challenges can be organized using a three-stage framework: pattern mining, pattern pruning and interactive visualization.

Index Terms—Temporal event sequence, visual analytics, frequent pattern mining, sequential data visualization and analysis

1 Introduction

Temporal event sequence data is pervasive in many application domains, including electronic commerce and digital marketing [10, 20], user workflow and behavior analysis [9, 19], online education [15, 17] and healthcare [12, 13]. Effective analysis of such data is challenging due to two main factors: the volume and complexity of the data, and the variety of user tasks and domain contexts. Real-world event sequences can be long and heterogeneous. The constituent events are multivariate and can have high cardinality [10, 12]. Traditional visualization and interaction techniques often fail to handle such data satisfactorily. Machine learning and data mining approaches are more scalable, but have paid relatively little attention to providing human-centered analysis methods and tools.

The ultimate solution to the problem of temporal and sequential event analysis will be a synthesis of research from relevant areas such as data management, data mining, visualization and human-computer interaction. In recent years, we have seen research works that seek to integrate sequential pattern mining with interactive visualization [8, 10, 13] and these approaches look promising. There are still many questions that remain unexplored and we take the position that these questions can be systematically organized in a framework consisting of three components: *pattern mining*, *pattern pruning* and *interactive visualization design*. Using this framework, we reflect on our experiences in working with event sequence data from two domains (web clickstreams and application logs from Adobe Photoshop), and discuss research questions at the intersection of human-centered sequence mining and visualization.

2 Pattern Mining: What Types of Patterns to Extract?

There are many different types of high-level structures we can extract from temporal event sequences. From a human-centered perspective, *interpretability* is a key measure when deciding which type to use. Model-based machine learning approaches such as Support Vector Machines are effective for prediction, but such models are usually difficult to understand for human users. Frequent pattern mining, on the other hand, offers a promising approach that provides easy-to-understand patterns for data exploration and analysis.

Research work on frequent pattern mining is extensive. Generally speaking, a taxonomy of frequent patterns has the following dimensions:

1. Order of Events When the extracted pattern is a set of events and the order of these events in the original sequences is not preserved, the pattern is an *itemset*. When we do preserve the order, the pattern is a *sequential pattern*.

2. Containment Two central concepts in frequent pattern mining are *support* and *containment* [1]. The support of a pattern is the number (or percentage) of input sequences matching that pattern. One pattern with a lower support may contain another pattern with a higher support. Given a threshold support *s*, a *frequent sequential pattern* is a sequence of events present in at least *s* input sequences. A *closed sequential pattern* is a frequent sequential pattern which includes as many events as possible without compromising the number of supported sequences. The definition of a *maximal sequential pattern* is even stricter: no other sequential patterns should contain a maximal pattern given *s*. Similarly, itemsets can be categorized as frequent itemsets, closed itemsets and maximal itemsets. Table 1 shows examples of these kinds of patterns.

3. Spatial Cohesion The distance between two adjacent events in a pattern may vary in the matching input sequences. At one extreme, an *n-gram* is a frequent sequential pattern consisting of contiguous events found in the input sequences. An *episode* is a frequent sequential pattern with events appearing compactly (within small window) in the matching sequences. A cohesive itemset [14] is similar except that the order of events is not enforced. In general, a frequent pattern does not have requirements on spatial cohesion.

Table 1: Examples of Frequent patterns

(a) Input Sequences

ID	Sequences
S1	ACD
S2	BCE
S3	EABC
S4	BE
S5	EBAC

(b) Patterns with >= 40% Support. For cohesive itemsets and episodes, we used the ratio of pattern length and window length as the cohesion parameter, and set the threshold value to 1.

Pattern Type	Patterns
Itemset	{A}, {B}, {C}, {E}, {A, B}, {A, C}, {A, E}, {B, C}, {B, E}, {C, E}, {A, B, C}, {A, B, E}, {A, C, E}, {B, C, E}, {A, B, C, E}
Closed Itemset	{C}, {A, C}, {B, E}, {B, C, E}, {A, B, C, E}
Maximal Itemset	{A, B, C, E}
Cohesive Itemset	{A, B}, {A, C}, {B, C}, {B, E}, {A, B, C}, {A, B, E}, {A, B, C, E}
Sequential Pattern	A, B, C, E, AC, BC, BE, EA, EB, EC
2-gram	BC
Episode	A, B, C, E, AC, BC

- *Zhicheng Liu, Mira Dontcheva and Matthew Hoffman are with Adobe Research. E-mail: {leoli,mirad,mathoffm}@adobe.com..*
- *Himel Dev is with Adobe Research and University of Illinois at Urbana Champaign. E-mail: hdev3@illinois.edu.*

Proceedings of the IEEE VIS 2016 Workshop on Temporal & Sequential Event Analysis. Available online at: http://eventevent.github.io

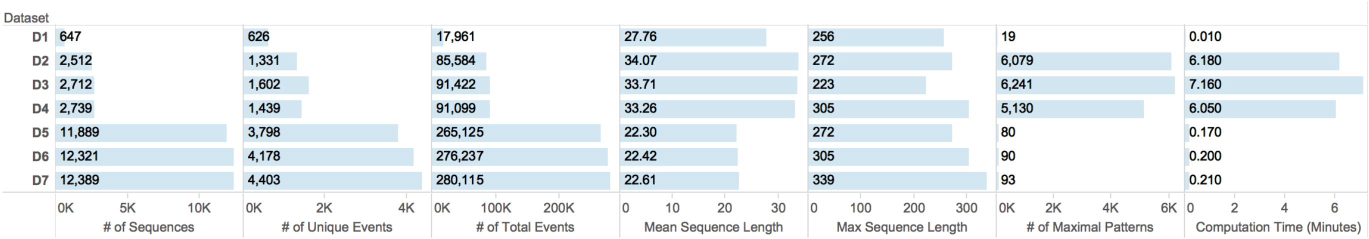

Fig. 1: Statistics on the datasets we have analyzed and the maximal sequential patterns extracted from these datasets. The patterns are computed on a quad-core 2.7 GHz MacBook Pro (OS X 10.11.5) with per-core 256K L2 caches, shared 6MB L3 cache and 16GB RAM.

In addition to these three dimensions, a pattern can also be periodic [5], multi-dimensional [6] or emerging across datasets [2].

With so many variations, the decision on what types of patterns to extract depends on two main factors: user task and domain context. When we worked with web clickstream analysts, their main task was to identify common visitor paths [10]. In this context, the order of pages navigated is important and must be preserved. As a result, we chose a type of sequential pattern: *maximal sequential pattern*. When we talked to product managers and machine learning scientists who work with Photoshop log data, however, their main concern was to break down event sequences into meaningful groups of actions that might correspond to actual Photoshop user tasks. Since an image editing task can be done in a variety of ways using the same set of operations, the order of events is not important. In addition, it does not make sense for the set of events in a user task to be spatially distributed across a long sequence. We thus chose to mine *spatially cohesive itemsets* in this case.

3 Pattern Pruning: What Patterns to Show?

Given that we have chosen a type of pattern to compute, the number of patterns found in a dataset can be overwhelming. For example, we computed maximal sequential patterns on seven different datasets in our previous work [10], and the results are shown in Figure 1. Even though maximal sequential pattern is a much stricter definition than other types of patterns such as frequent itemsets, the number of patterns extracted can be greater than the input sequences for some datasets. Presenting all these patterns to the users is not an advisable idea.

To prune the number of patterns down to a manageable size, various techniques can be used. Peekquence [8] allows users to sort and filter patterns based on pattern attributes such as pattern length and event variability. Automatic approaches that discard patterns based on overlap between sequences in the support sets have also been proposed [10].

Ultimately, we would like to present our users patterns that are interesting and insightful. Based on our conversation with analysts working on web clickstream data as well as log data, several characteristics stand out as important measures of interestingness. First, longer patterns are usually considered more interesting, because they contain more information. Second, the size of the input sequence dataset plays a role. When computing sequential patterns for a large dataset spanning days for all the visitors, the extracted patterns are often not very interesting. However, when we segment the dataset by user groups or other meaningful attributes, interesting patterns begin to surface. This observation is contrary to the belief that we should start with an overview of the entire dataset. Carefully segmenting the dataset seems a prerequisite to effective pattern mining. Finally, the length of input sequences affects the interestingness of mined pattern as well. Some Photoshop users leave their applications open for days. If we group the events by user ID, the resulting sequences can be extremely long and the frequent patterns are not that meaningful. When we divide a user sequence into sessions using timestamps, the extracted patterns tend to make more sense.

Since the concept of *interestingness* is central to the question “What patterns should we show to the users?”, we need a systematic approach that tries to quantify and predict the interestingness of sequential patterns. Related previous work has explored this problem from a data mining perspective. Freitas argues that the term interestingness is related to several properties such as novelty, surprisingness and usefulness [3]. Silberschatz and Tuzhilin note that “a pattern is interesting when the user can do something about it” [18]. In other words, interestingness and actionability are often related. Measures of interestingness may be objective or subjective [3]. Objective approaches try to characterize interestingness as a formal, inherent property of the pattern itself. Examples of objective measures are often based on information theory. In comparison, subjective measures of interestingness assume that the interestingness of a pattern is relative to a belief system [18]. An expert user may find less patterns to be surprising than a novice.

Applying these concepts to temporal event sequence analysis, we face the research questions of devising objective and subjective measures of interestingness, surprisingness and actionability for frequent patterns. In particular, a data-driven approach to modeling and predicting interestingness seems promising. For a given set of computed patterns, we can ask users to rate the interestingness of each pattern, and build models that use pattern features such as event names and variability to predict interestingness scores. It might make sense to group users by their expertise so that we take into consideration of their background and prior knowledge.

4 Interactive Visualization: How to Show Patterns and Sequences?

Interactive visualizations of frequent patterns and input sequences have potentially two goals: to understand and debug frequent pattern mining algorithms, and to analyze temporal event sequence datasets for knowledge discovery. Most of the visual analytic work in this space seems to be focusing on the second use case.

Research work on sequence visualization is extensive, investigating visual representations ranging from flow diagrams [4, 21] to tree visualizations [12, 22]. When faced with large, high-dimensional datasets, users can start with a complex visualization and simplify it by specifying patterns such as motifs and subsequences [12]. Alternatively, users can also start with an empty canvas and incrementally build visualizations by querying patterns in terms of milestone events [4, 9, 23]. In these works, patterns are specified by users.

When patterns are automatically mined, we need visualizations that explain the relationship between the patterns and sequences. Depending on the type and properties of frequent patterns, we may need to design different visual representations for the extracted patterns. In Peekquence [8] and in our work [10], individual patterns are visualized as separate entities, and it is not very clear how these patterns relate to each other and fit into a larger picture with the input sequence dataset in mind. Fp-Viz [7] uses a radial layout to show the hierarchy of frequent itemsets and this approach may be extended further to organize frequent patterns in a meaningful overview.

The necessity of visualizing individual sequences in conjunction with frequent patterns remains unclear. While it is nice to be able to drill down to individual input sequences and examine the raw data, often the input dataset is too huge for every input sequence to be displayed. Sampling techniques for sequences can be useful to decide what individual sequences to show, if any. Exploring use cases where the coordination between patterns and sequences seems an interesting problem to be pursued.

Ultimately, the analysis of temporal event sequence should enable analysts to answer domain-specific questions they care about. For example, web clickstream analysts are interested in understanding the most common paths. In doing so, they hope to understand if there is an anomaly in the traffic flow, where do visitors drop off, and how to guide them into a funnel. In the case of anomaly detection, data becomes more meaningful when the analysts can *compare* the traffic pattern across different time periods and user segments. Event sequence visualization for comparison has been explored in the past [11, 24], but not extensively in the context of frequent pattern mining.

Finally, while frequent patterns serve as a useful summary of the dataset, some of these questions may not be adequately answered by the visualization. To understand *why* a pattern is present in the dataset, we may need to collect additional datasets that are orthogonal but complementary to the input temporal event sequences.

5 Conclusion

To support effective analysis of temporal event sequence data, researchers and developers of analytic systems need to make informed design decisions during the pattern mining, pattern pruning and visualization design stages. We discuss research challenges and opportunities in each of the stages. This framework is primarily useful for thinking about issues in combining frequent pattern mining algorithms with visualization interfaces. There are other interesting approaches to temporal event sequence analysis that we have not addressed in this paper. For example, research on visualization design might lead to new approaches on human-centered frequent pattern mining. Visualizations can be combined with machine learning techniques [16] to support mixed-initiative interfaces for semi-automatic segmentation of event sequences. The three-stage framework is thus subject to further refinement when new research questions arise.

References

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pp. 3–14. IEEE, 1995.

[2] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 43–52. ACM, 1999.

[3] A. A. Freitas. On objective measures of rule surprisingness. In *European Symposium on Principles of Data Mining and Knowledge Discovery*, pp. 1–9. Springer, 1998.

[4] D. Gotz and H. Stavropoulos. Decisionflow: Visual analytics for high-dimensional temporal event sequence data. *IEEE transactions on visualization and computer graphics*, 20(12):1783–1792, 2014.

[5] J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pp. 106–115, Mar 1999. doi: 10.1109/ICDE.1999.754913

[6] M. Kamber, J. Han, and J. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. In *KDD*, vol. 97, p. 207, 1997.

[7] D. A. Keim, J. Schneidewind, and M. Sips. Fp-viz: Visual frequent pattern mining. 2005.

[8] B. C. Kwon, J. Verma, and A. Perer. Peekquence: Visual analytics for event sequence data. In *ACM SIGKDD 2016 Workshop on Interactive Data Exploration and Analytics*, 2016.

[9] H. Lam, D. Russell, D. Tang, and T. Munzner. Session viewer: Visual exploratory analysis of web session logs. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pp. 147–154, Oct 2007. doi: 10.1109/VAST.2007.4389008

[10] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics*, 23(01), Janaury 2017.

[11] S. Malik, F. Du, M. Monroe, E. Onukwugha, C. Plaisant, and B. Shneiderman. Cohort comparison of event sequences with balanced integration of visual analytics and statistics. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pp. 38–49. ACM, New York, NY, USA, 2015. doi: 10.1145/2678025.2701407

[12] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal event sequence simplification. *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2227–2236, 2013.

[13] A. Perer and F. Wang. Frequence: interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pp. 153–162. ACM, 2014.

[14] C. Pierre. A new constraint for mining sets in sequences. 2009.

[15] H. Qu and Q. Chen. Visual analytics for mooc data. *IEEE Computer Graphics and Applications*, 35(6):69–75, Nov 2015. doi: 10.1109/MCG.2015.137

[16] A. Saeedi, M. D. Hoffman, M. J. Johnson, and R. P. Adams. The segmented ihmm: A simple, efficient hierarchical infinite hmm. In *ICML*, 2016.

[17] C. Shi, S. Fu, Q. Chen, and H. Qu. Vismooc: Visualizing video clickstream data from massive open online courses. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 159–166. IEEE, 2015.

[18] A. Silberschatz and A. Tuzhilin. On subjective measures of interestingness in knowledge discovery. In *KDD*, vol. 95, pp. 275–281, 1995.

[19] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao. Unsupervised clickstream clustering for user behavior analysis. In *SIGCHI Conference on Human Factors in Computing Systems*, 2016.

[20] J. Wei, Z. Shen, N. Sundaresan, and K.-L. Ma. Visual cluster exploration of web clickstream data. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pp. 3–12. IEEE, 2012.

[21] K. Wongsuphasawat and D. Gotz. Outflow: Visualizing patient flow by symptoms and outcome. In *IEEE VisWeek Workshop on Visual Analytics in Healthcare, Providence, Rhode Island, USA*, pp. 25–28. American Medical Informatics Association, 2011.

[22] K. Wongsuphasawat and J. Lin. Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at twitter. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pp. 113–122. IEEE, 2014.

[23] E. Zgraggen, S. M. Drucker, and D. Fisher. (s—qu)eries: Visual regular expressions for querying and exploring event sequences. *Proceedings of CHI 2015*, 2015.

[24] J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. Matrixwave: Visual comparison of event sequence data. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 259–268. ACM, 2015.