EuroVis Workshop on Visual Analytics (2020)
K. Vrotsou and C. Turkay (Editors)

A window-based approach for mining long duration event-sequences

Katerina Vrotsou and Aida Nordman

Linköping University, Media and Information Technology, Department of Science and Technology, Sweden

Abstract

This paper presents an interactive sequence mining approach for exploring long duration event-sequences and identifying interesting patterns within them. The approach extends previous work on exploratory sequence mining by using a sliding window to split the sequence prior to mining. Patterns are interactively grown and visualized through a tree representation, while a set of accompanying views allows for identified patterns to be explored in the context in which they occur. The approach is motivated and exemplified in the domain of air traffic control and, in particular, air traffic controller training.

CCS Concepts

• ***Human-centered computing*** → ***Visual analytics;*** • ***Information systems*** → ***Data mining;***

1. Introduction

Event-sequence data are today found in numerous and diverging domains such as healthcare, process control, and eye tracking. The composition of the datasets can vary dramatically across domains, in terms of types of events, size of event alfabet and sequence length. The task of interest, however, is commonly concerned with the identification of patterns defined as sequences displaying some interesting behaviour. For example, the sequence of symptoms following a certain treatment, events leading up to an alarm, or actions for performing given tasks. Given the nature of the data and the intended analysis, sequential pattern mining has often been proposed as an appropriate analysis method to identify such patterns.

Sequential pattern mining addresses the problem of identifying sub-sequences of events as patterns in large event-sequence datasets. The common general approach to do this is to create candidate patterns and test their support against a user-specified threshold. This is done by, the candidate patterns are sought in the data and when a match is found in an event-sequence the support is increased and the next event-sequence is controlled. This traditional approach works best for short(er) sequences, where frequency of patterns is computed *between* event-sequences; this implies that a pattern is frequent if it occurs in many sequences. However, in domains dealing with fewer sequences that have, instead, very long duration it is often interesting to compute frequency *within* event-sequences; implying that a pattern is frequent if it occurs repeatedly within a single sequence. Gaze analysis made possible through eye-tracking data is one such domain.

Eye-tracking data are commonly sampled with high frame rates and collected over long periods of time (hours), resulting in long and detailed sequences. Their collection is interesting in many monitoring tasks such as studying attention patterns of drivers, monitoring large process control systems, and understanding activity patterns when training on a simulator. This last area is where we focus our work. This paper presents an interactive mining approach for exploring sequential patterns in long event-sequences. The approach expands previous work on exploratory sequence mining based on pattern-growth [VN19] and complements it with an extended windowing-based algorithm adjusted for the specific pattern identification needs arising from very long sequences. Our work is motivated in the area of air traffic control and finds application on event-sequences extracted from eye-tracking data during air traffic controller training, as described in the following section.

2. Motivation and domain characterisation

The targeted domain and motivation behind this work is Air Traffic Control (ATC); specifically air traffic controller (ATCO) training. A current challenge faced today in ATC is concerned with how to formalise the process of training ATCOs. ATCO training is primarily performed using simulators, where ATCOs successively practise ATC scenarios of increasing complexity, while an instructor is observing them. Their performance is followed up by discussing the choices made during the scenario. ATCO training is commonly described in the ATC community as a handicraft because there are not many predefined scan patterns, or formalised instructions, as to how an ATCO is expected to scan their environment and perform relevant tasks. Thus, there is an increased interest for approaches that can improve the process and effectiveness of ATCO training.

One way to better understand the visual scan patterns of ATCOs in training is to collect eye-tracking data while they are performing scenarios on the simulator and use it as support in the discussion between trainee and instructor. Understanding, however, what the trainee is looking at remains a challenge for the instructor because

DOI: 10.2312/eurova.20201087

(1) attention shifts rapidly and scanning tasks occur in seconds so it can be difficult to detect them "live", and (2) as mentioned, there are not well-established formal scan patterns available that the instructor can look for in the data. To this end, it becomes interesting to explicitly identify which specific regions in the field of view (FOV) of the ATCO have received attention and in which order, as well as to be able to make assessments as to the correctness of these interactions; i.e. whether the ATCO has looked at the "correct" regions in the "correct" order. Since the knowledge of what is correct lies with the instructor, analysis methods need to incorporate input from the domain expert. This is where we make our contribution.

Raw gaze data collected through eye tracking experiments can be converted into discrete event-sequences by first dividing the FOV of the ATCO into distinct Areas Of Interest (AOI) and then aggregating the data into sequences of visits to these AOIs; i.e. event-sequences. Interactive sequence mining techniques could then be applied in order to identify specific sub-sequences of AOI visits as patterns. However, since in this domain setting, what is of primary interest is to detect individual scan patterns of different ATCOs, the mining process needs to be adjusted for seeking patterns within event-sequences rather than between them. To this end, we propose an approach that combines the exploratory pattern-growth based approach presented in [VN19] with the use of a sliding window to split the input sequence and a visualization that allows the exploration of the identified pattern events. The resulting approach facilitates the analysis of long sequences of events.

3. Related work

A large body of work has focused specifically on the visualization of eye-tracking data and the AOI related challenges this brings [BKW13, BKR*16, BKR*17]. Even though our work is motivated by the ATC domain and is exemplified by AOI sequences collected through eye-tracking experiments, our main focus is the exploration of sequential patterns in long duration event-sequences.

A lot of work has focused on exploring event-sequences in search of patterns. Several proposed approaches focus on creating appropriate representations of the event-sequences so that patterns can be visually identified in the data. In doing so, approaches have used, for example, alignment [WPQ*08], simplification [MLL*13], clustering [GXZ*17], and aggregation [WG12, CW18]. An interesting summary of analysis strategies can be found in [DSP*17]. The use of sequence mining has also gained popularity in the exploration of patterns in event-sequences. The use of various algorithms has been proposed in combination with representations offering flexible exploration of the resulting patterns [PW14, PCK*16, LKD*17]. There are also several examples of approaches that integrate users' domain knowledge into the mining process, attempting in this way more focused and relevant results [SPG14, LLMB19, VN19].

Most proposed visual analysis approaches focus on identifying common patterns between event-sequences. In many practical problems, such as mining a sequence from eye-tracking experiments, it is important to detect typical behaviour exhibited by the person performing the experiment. So, what is interesting in such a situation is to identify repeating patterns within single (or few) long sequences. To address this problem we propose the use of a sliding window in order to split the input sequence. The concept of sliding windows is not new in data mining, it appears with different uses in the domains of sequence and data stream mining [RS02, GZK05]. In sequence mining sliding time windows are commonly used for assigning temporal constraints; i.e. a time interval within which a pattern should occur [SA95, YZMKR16]. In data stream mining sliding time windows are commonly used for mining the most recent patterns [TAJL09, NDSZ11]. Also, similar to time series analysis, there have been examples of using the notion of time windows in order to segment sequence data prior to mining [CSD98, DLM*98], which is where we base our approach.

4. Method description

The method presented in this paper builds further on a previously proposed interactive pattern-growth based mining algorithm implemented within the ELOQUENCE system [VN19]. The most distinctive feature of this algorithm is the possibility to interactively add local constraints on the patterns being built in order to increase their focus. Different types of constraints are supported in the approach; including *event constraints*, *gap constraints* and *data filters*.

4.1. Algorithm

The work described here extends the authors' previous work by facilitating the analysis of single or few long sequences of events. To achieve this, a time window is used which slides over the input sequence, one unit of time in each step. In other words, one can imagine a series of time windows, all with the same time span, and having one time unit's difference in their respective start and end times. The size of the sliding window, denoted as W_D, is set by the analyst. In the following the algorithm is discussed for single long sequences but could in the same manner be applied to several sequences. The number of windows would then be computed with respect to the summed duration of the event sequences.

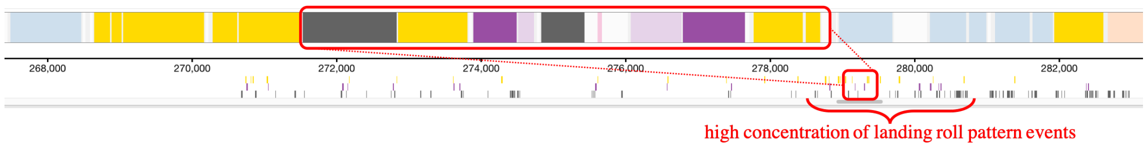
Let T_E be the ending time of the input sequence and assume the start time is zero ($T_0 = 0$). Then, the number of consecutive windows over the input sequence is given by $T_E + W_D - 1$. The support of a sequence S is then defined as the fraction of the total number of consecutive windows, where S occurs, and S is a pattern if its support is above a given threshold $\sigma > 0$ i.e.

$$\frac{\text{number of windows of size } W_D\text{, where } S \text{ occurs}}{T_E + W_D - 1} \geq \sigma.$$

Through an initial pre-processing step, $T_E + W_D - 1$ subsequences can be generated from the input sequence, each corresponding to one consecutive time window. Then, the pattern-growth based mining algorithm of ELOQUENCE [VN19] can be used to explore patterns. Since patterns must be observed within time windows of a given duration, patterns can have at most as many events as the windows duration. In practice this does not need to be a limitation because, in most practical applications, meaningful patterns are composed of events occurring in close proximity. The definition of closeness depends on the application area and is reflected in the time unit that the sequences are expressed in. In the current ATC example, time is expressed in milliseconds and events

(a) Instance of an initial approach pattern (RADAR_RUNWAY→AWOS_QNH) identified in the event sequence.

(b) Instance of a runway scan pattern (OTW_APPROACH→OTW_RUNWAY_1L→OTW_RUNWAY_2L) identified in the event sequence.



(c) Instance of a landing roll pattern (OTW_APRON→OTW_RUNWAY_4R→OTW_RUNWAY_3C) identified in the event sequence.

Figure 1: *Examples of three identified scan patterns highlighted in context in the sequence view and event overview.*

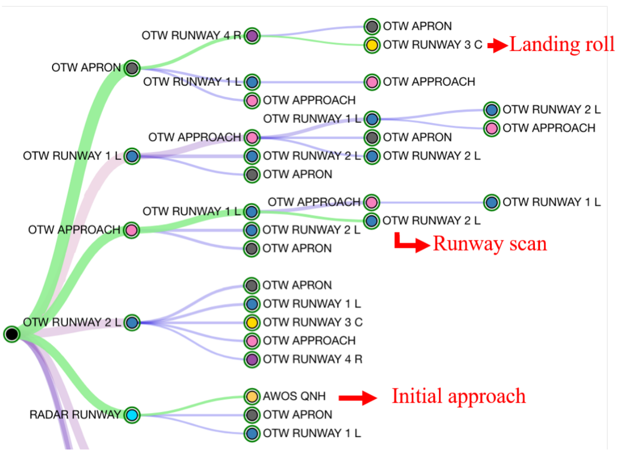


Figure 2: *Pattern tree with example pattern sequences highlighted.*

belonging to the same pattern are expected to occur within a few seconds from each other, which reflects the duration of tasks performed during ATC scenarios.

Finally, if one assumes that each event's dwell time is one unit of time then the window's size can also be defined in terms of number of events, instead of a duration (i.e., events' dwell time is ignored).

4.2. User interface

The proposed approach is implemented as part of the exploratory visual sequence mining system, ELOQUENCE [VN19]. The previous system has been extended with the new algorithm, described in section 4.1, and a new event overview representation. The three main views of the extended interface include (1) the pattern tree view, (2) the sequence view, and (3) the event overview. Moreover, a panel is available where a user can interactively set constraints in order to tailor the mining process as described in [VN19]. The views, described shortly below, are illustrated in Figures 2 and 1.

1. The *pattern tree view* is a tree representation used to interactively drive the sequence mining algorithm and display the patterns as they are mined. Sequences of interest are grown by stepwise clicking on the nodes of the tree (Figure 2).
2. In the *sequence view* sequences are displayed as horizontal bars comprising the different events (e.g. Figure 1(a) top part). The user can choose to display the original sequences composing the dataset, or the split sequences on which the mining is being performed; i.e. the sequences in the time windows.
3. The *event overview* is a representation of the distribution of pattern events across the event sequence. Timelines of each event of a selected pattern are stacked on top of each other providing an overview of their frequency of occurrence and their distribution (e.g. Figure 1(a) bottom part). The event overview depicts thus, in a single view, how a selected pattern's events are spread across the entire event sequence, while the sequence view only depicts a selected portion at a time. The time axes in both views show relative time from the beginning of the event sequence (i.e. $t_0 = 0$). Instances of a pattern selected in the pattern tree view can be identified in the event overview by inspecting the positions where the events composing the pattern occur in close proximity to each other and (preferably) in the same order.

All views of ELOQUENCE are linked. Colour reflects the event type, here the AOI (Figure 3). Selecting a pattern in the pattern tree view will highlight the pattern events in the *sequence view* (Figure 1(a) top) and the *event overview* (Figure 1(a) bottom). Clicking on an event in the event overview places that event in focus by centring it in the sequence view.

5. Use cases

We have applied our approach to an eye-tracking dataset composed of ATCOs performing landing and take-off scenarios on a simulator. The data were collected using TOBII Pro Glasses 2 head-mounted glasses with a sampling frequency of 50Hz. Following this, the continuous gaze data were translated to discrete sequences of AOI visited by the ATCO when performing the scenarios. To extract such sequences, a division of the FOV for the reference scenarios into 24 unique AOIs was created by an instructor (Figure 3).

Examples of patterns identified from exploring a single landing scenario are discussed below. The mining algorithm was run with the following settings. The size of the sliding window was set to $W_D = 100$ *seconds*. The maximum gap allowed between pattern events was set to 10 events, and the maximum time between pattern events was set to 10 *seconds*. The exploration of the data was initiated with a minimum support of 30% and the pattern tree was expanded one level. Following this, the minimum support was reduced to 10% and all branches were fully expanded (Figure 2).

The following three central patterns could be identified from the pattern tree which correspond well to template scan patterns identified by ATCO instructors in previous work [WVN*19].

1. **Initial approach pattern**. The pattern is composed of the AOI sequence `RADAR_RUNWAY→AWOS_QNH` (Figure 2). The pattern is characterised by the ATCO switching their attention between the radar monitor, where they are presumably following the aircraft as it is approaching the airport, and the Automated Weather Observation System (AWOS), where they are monitoring weather conditions. Highlighting the events composing this pattern in the event overview reveals that these are occurring primarily in the beginning of the landing scenario before the ATCO has made visual contact with the aircraft (Figure 1(a)).
2. **Runway scan pattern**. The pattern is composed of the AOI sequence `OTW_APPROACH→OTW_RUNWAY_1L→OTW_RUNWAY_2L` (Figure 2). Here the ATCO is repeatedly looking at the Out The Window (OTW) approach area anticipating visual contact with the aircraft and then checking the runway for obstacles, by scanning across the runway AOIs, to give landing clearance. Inspecting the pattern in the event overview shows that runway scans appear more frequently after the ATCO has made visual contact with the aircraft and until the aircraft has landed (Figure 1(b)).
3. **Landing roll pattern**. The pattern is composed of the AOI sequence `OTW_APRON→OTW_RUNWAY_4R→OTW_RUNWAY_3C` (Figure 2). The ATCO is inspecting the apron for obstacles and then follows the aircraft slowly moving from the right part of the runway back towards the middle part where the taxiway into the apron is placed (Figure 3). The event overview reveals that the pattern takes over after the aircraft has landed and repeats until it has entered the apron and stops moving (Figure 1(c)).

6. Conclusions

We have proposed an interactive sequence mining approach that allows an analyst to explore long duration event-sequences in search of patterns within them. The approach extends previous work on exploratory sequence mining [VN19] by adding an algorithm that uses a sliding time window to split the event-sequences and then mines patterns from the time windows. Patterns are interactively grown and visualized through a tree representation and a set of accompanying views allows for identified patterns to be explored in the context in which they occur. The presented approach is motivated in the domain of air traffic control, and in particular by ATCO training. The objective has been to create an interactive tool that can assist an instructor in identifying ATCOs' visual scan patterns, while training on ATC scenarios. Such a tool has the potential to be used to assess whether tasks are performed "as expected" and form a base for discussion between instructor and trainee.

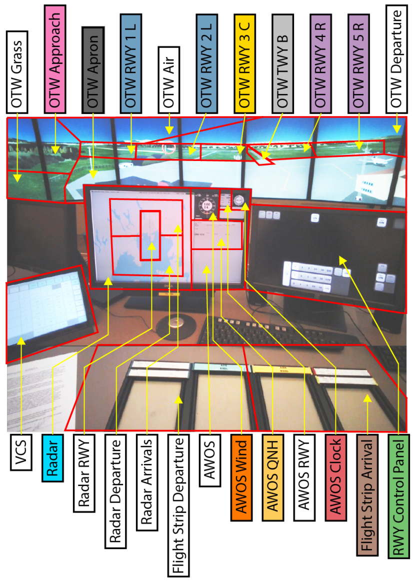


Figure 3: *The 24 unique AOIs in the FOV. Abbreviations in the figure: Automated Weather Observing System (AWOS), Voice Communication System (VCS), Out The Window (OTW), Runway Control Panel (RCP), Runway (RWY), QNH refers to atmospheric pressure.*

A limitation of the proposed work is that the chosen time window's size (W_D) has a direct effect on the patterns that can be mined; patterns longer than the window can never be mined. This puts responsibility on the analyst to choose the setting appropriately and highlights the importance of including domain knowledge in the process. Moreover, the approach has primarily focused on mining patterns from single long sequences. As as consequence, the accompanying event overview was designed to reveal the distribution and ordering of pattern events along a time line and only works on single sequences. Finally, variations in the ordering of pattern events are not explicitly identified using the algorithm. In the particular domain of focus this can be a limitation since the order in which regions are scanned by ATCOs is not necessarily strict. Alternative approaches, based for example on regular expressions, which can handle this are interesting to explore as future work.

Other future work of interest is to test the proposed window-based algorithm with larger datasets, comprising several ATCO training sessions, in search of common behaviours between ATCOs. This could indicate whether there are in fact "common ways" of doing tasks and create the basis for an "instruction book" with more explicitly formalised ATC visual scanning patterns.

References

[BKR*16] Blascheck T., Kurzhals K., Raschke M., Strohmaier S., Weiskopf D., Ertl T.: AOI hierarchies for visual exploration of fixation sequences. *Eye Tracking Research and Applications Symposium (ETRA) 14* (2016), 111–118. 2

[BKR*17] Blascheck T., Kurzhals K., Raschke M., Burch M., Weiskopf D., Ertl T.: Visualization of Eye Tracking Data: A Taxonomy and Survey. *Computer Graphics Forum 36*, 8 (2017), 260–284. 2

[BKW13] Burch M., Kull A., Weiskopf D.: AOI rivers for visualizing dynamic eye gaze frequencies. *Computer Graphics Forum 32*, 3 PART3 (2013), 281–290. 2

[CSD98] Chakrabarti S., Sarawagi S., Dom B.: Mining surprising patterns using temporal description length. In *24th VLDB Conference* (1998), vol. 98, pp. 606–617. 2

[CW18] Cappers B. C. M., Wijk J. J. V.: Exploring Multivariate Event Sequences using Rules , Aggregations , and Selections. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 532–541. 2

[DLM*98] Das G., Lin K.-I., Mannila H., Renganathan G., Smyth P.: Rule Discovery from Time Series. In *KDD* (1998), American Association for Artificial Intelligence. 2

[DSP*17] Du F., Shneiderman B., Plaisant C., Malik S., Perer A.: Coping with volume and variety in temporal event sequences: Strategies for sharpening analytic focus. *IEEE Transactions on Visualization and Computer Graphics 23*, 6 (2017), 1636–1649. 2

[GXZ*17] Guo S., Xu K., Zhao R., Gotz D., Zha H., Cao N.: EventThread: Visual Summarization and Stage Analysis of Event Sequence Data. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 56–65. 2

[GZK05] Gaber M. M., Zaslavsky A., Krishnaswamy S.: Mining data streams: A review. *SIGMOD Record 34*, 2 (2005), 18–26. 2

[LKD*17] Liu Z., Kerr B., Dontcheva M., Grover J., Hoffman M., Wilson A.: CoreFlow: Extracting and Visualizing Branching Patterns from Event Sequences. *Computer Graphics Forum 36*, 3 (2017), 527–538. 2

[LLMB19] Law P. M., Liu Z., Malik S., Basole R. C.: MAQUI: Interweaving queries and pattern mining for recursive event sequence exploration. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2019), 396–406. 2

[MLL*13] Monroe M., Lan R., Lee H., Plaisant C., Shneiderman B.: Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (dec 2013), 2227–36. 2

[NDSZ11] Nori F., Deypir M., Sadreddini M. H., Ziarati K.: A new sliding window based algorithm for frequent closed itemset mining over data streams. *2011 1st International eConference on Computer and Knowledge Engineering, ICCKE 2011* (2011), 249–253. 2

[PCK*16] Polack P. J., Chen S.-T., Kahng M., de Barbaro K., Sharmin M., Basole R., Chau D. H.: Chronodes: Interactive Multifocus Exploration of Event Sequences. *CoRR abs/1609.0* (2016). 2

[PW14] Perer A., Wang F.: Frequence : Interactive Mining and Visualization of Temporal Frequent Event Sequences. In *International Conference on Intelligent User Interfaces* (Haifa, Israel, 2014), ACM, pp. 153–162. 2

[RS02] Roddick J. F., Spiliopoulou M.: A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering 14*, 4 (2002), 750–767. 2

[SA95] Srikant R., Agrawal R.: *Mining Sequential Patterns: Generalizations and Performance Improvements*. Tech. rep., IBM Almaden Research Center, San Jose, California, 1995. 2

[SPG14] Stolper C. D., Perer A., Gotz D.: Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1653–1662. 2

[TAJL09] Tanbeer S. K., Ahmed C. F., Jeong B. S., Lee Y. K.: Sliding window-based frequent pattern mining over data streams. *Information Sciences 179*, 22 (2009), 3843–3865. 2

[VN19] Vrotsou K., Nordman A.: Exploratory Visual Sequence Mining Based on Pattern-Growth. *IEEE Transactions on Visualization and Computer Graphics 25*, 8 (2019), 2597–2610. 1, 2, 3, 4

[WG12] Wongsuphasawat K., Gotz D.: Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization. *IEEE Transactions on Visualization and Computer Graphics 18*, 12 (dec 2012), 2659–2668. 2

[WPQ*08] Wang T. D., Plaisant C., Quinn A. J., Stanchak R., Murphy S.: Aligning Temporal Data by Sentinel Events : Discovering Patterns in Electronic Health Records. *CHI 2008 Proceedings · Health and Wellness* (2008), 457–466. 2

[WVN*19] Westin C. A., Vrotsou K., Nordman A., Lundberg J., Meyer L.: Visual scan patterns in tower control: Foundations for an instructor support tool. *SESAR Innovation Days*, December (2019), 1–8. 4

[YZMKR16] Yusof N., Zurita-Milla R., Kraak M. J., Retsios B.: Interactive discovery of sequential patterns in time series of wind data. *International Journal of Geographical Information Science 30*, 8 (2016), 1486–1506. 2