Spatiotemporal event sequence discovery without thresholds

Berkay Aydin¹ · Soukaina Filali Boubrahimi¹ · Ahmet Kucuk¹ · Bita Nezamdoust¹ · Rafal A. Angryk¹

Received: 17 December 2019 / Revised: 1 September 2020 / Accepted: 22 October 2020 /

Published online: 9 November 2020

© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Spatiotemporal event sequences (STESs) are the ordered series of event types whose instances frequently follow each other in time and are located close-by. An STES is a spatiotemporal frequent pattern type, which is discovered from moving region objects whose polygon-based locations continuously evolve over time. Previous studies on STES mining require significance and prevalence thresholds for the discovery, which is usually unknown to domain experts. The quality of the discovered sequences is of great importance to the domain experts who use these algorithms. We introduce a novel algorithm to find the most relevant STESs without threshold values. We tested the relevance and performance of our threshold-free algorithm with a case study on solar event metadata, and compared the results with the previous STES mining algorithms.

Keywords Spatiotemporal data mining · Sequence patterns · Pattern mining

1 Introduction

In traditional itemset mining, frequent sequence (or sequential pattern) mining refers to discovering a set of attributes persistently appearing over time among the large number of objects [50]. A major category of sequences are event sequences, which represent the implicit relations among the categories of objects [16]. Classical event sequence mining can be useful for understanding the user behavior (by mining sequences from weblogs or system traces) [37], shopping routines of customers (by mining transaction sequences) [44], or the efficiency of business processes (by mining time-ordered managerial and operational activities) [52].

Sequential pattern mining from spatiotemporal data has received much attention in recent years due to its broad application domains such as targeted advertising, prediction for taxi services, and urban planning [15, 29, 40, 53]. The characteristics of spatiotemporal

✉ Berkay Aydin
baydin2@cs.gsu.edu

¹ Georgia State University, Atlanta, GA, USA

sequences vary widely depending on the discovered knowledge type. Most of the recent approaches focus on the point-based spatiotemporal data presumably due to its availability. However, the region-based spatiotemporal data, primarily obtained from scientific resources, has not received much attention. In this work, we focus on spatiotemporal event sequences (STES) from event datasets that contain instances with region-based geometric representations.

STESs are the ordered series of event types whose instances frequently demonstrate sequence generating behavior. The sequence generating behavior is characterized by the spatiotemporal *follow* relationship among instances, and it refers to the temporal follow relationship with spatial proximity constraints. The spatiotemporal sequences can be categorized into three classes based on the fundamental data type: sequences of trajectories from uniform groups, sequences of spatiotemporal points from mixed groups, and sequences of trajectories from mixed groups. Our work considers the last category, and we mine the STESs from event instances formed by the evolving region trajectories.

1.1 Motivation

The discovery of STESs is potentially critical for the large-scale verification and prediction of scientific phenomena in a broad range of scientific fields including meteorology, geophysics, epidemiology, and astronomy [18]. The scientific phenomena such as tornadoes, propagation of epidemics, clouds, and solar events can be modeled as trajectories of continuously evolving regions. STES mining can be used for modeling the spatial and temporal relationships among different types of phenomena. Later, the discovered sequence patterns can be utilized for performing large-scale verification of current knowledge, as well as the prediction of unknown spatiotemporal relationships among different event types.

Real-life examples An application area for STES mining is solar physics and space weather forecasting. Studies from government agencies [10, 28] and independent institutions [30, 39] show that extreme space weather events can impact radiation in space, reduce the safety of space and air travel, disrupt intercontinental communication and GPS, and even damage power grids. While much work has been done on prediction of solar flares using physical characteristics of source active regions, the mixed impact of different types of solar events to eruptive activity (such as flares or coronal mass ejections) have not been fully explored. For example, in Fig. 1, we demonstrate an active region event with a large sunspot followed by a flare. Such relationships are known to exist among solar events [3, 25, 32, 36] although, to our knowledge, many studies are confined to a limited number of examples. One way to understand the exhaustive factors and conditions leading to an extreme space weather event

Fig. 1 An active region and a sunspot instance, occurred on January 20, 2012, followed by an instance of a solar flare. (The background images are obtained from Helioviewer.org (AIA 171 Angstrom) and are a courtesy of SDO/AIA team)

is to determine the frequently occurring sequences of events which lead to substantially large flares, smaller flares, and non-flaring instances. The discriminating event sequences among these can shed light into typical conditions leading to eruptions or alternatively help forecasters when issuing all-clear forecasts.

Public health researchers, particularly epidemiologists, can also benefit from STES mining for understanding the frequently occurring activities leading to spread of infectious diseases. Contact tracing applications [1] are one of the very few tools we can deploy to stop the spread of viral diseases with great epidemic potential such as the novel coronavirus (SARS-CoV-2) [19]. Such applications can be used to trace the movements of individuals and the paths of individuals can be split into activity event types. Mining STESs occurring among different activities in the outbreak zones can help epidemiologists understand which activities or the sequences of activities lead to outbreaks and which are relatively safer. Identifying these sequences can provide crucial information for prevention such as the relative contributions of different activities and ways of transmission.

1.2 Contributions

Previous efforts have been devoted to solving the problem of mining the most prevalent spatiotemporal event sequences using Apriori [5], pattern growth-based [4], or Top-K algorithms [9]. While these three approaches achieve the expected results, they heavily rely on user-defined significance and prevalence threshold parameters, which define the cut-off points for sequences. The previous algorithms assume that the user has a prior knowledge of the optimal threshold parameters or a K value which in some cases, should be discovered from the datasets.

Another issue that surface with the previous STES mining approaches is that the prevalence threshold, is highly dependent on the events taking part in a sequence. For example, algorithms should be more tolerant in the case of STESs whose event types have a rare occurrence, yet it should also be informative on the rareness of them. As a matter of fact, defining the same threshold for all the sequences may not be accurate in this context as the threshold parameter should discriminate against the event types participating in a given event sequence.

In contrast to threshold-based approaches, we focus on overcoming the limitations of providing a user-defined threshold when discovering the STESs and improve the relevance of our results. Here, we introduce a novel algorithm, RAND-ESMINER, which, by randomly repeating the mining process on a random subset of instances and *follow* relationships, finds an estimate participation index for event sequences. The RAND-ESMINER uses our pattern growth-based ESGROWTH algorithm [4] as the backbone, where the *follow* relationships are translated into a directed acyclic graph structure, and randomly permutes the edges of this graph to mine the event sequences. Mining the random subset multiple times does not necessarily improve the overall running time as shown in our experiments; however, it increases the robustness of the results and let us understand the distribution of participation indices. In our experiments, we compare the RAND-ESMINER with earlier approaches and evaluate the efficiency and robustness of the algorithm using datasets from solar astronomy field.

1.3 Related work

In the spatiotemporal frequent pattern mining literature, the term *sequence* (or its derivatives such as sequence patterns, sequential patterns) is used for identifying different types

of knowledge from spatiotemporal data. These include sequences of locations frequently visited by spatiotemporal objects [14, 20], partially- or totally-ordered sequences of event types whose instances follow each other [5, 13, 24, 34, 35, 49] (these are also referred to as couplings in [42]), sequences of semantic annotations from semantic trajectories [51], temporal sequences of ordered spatial itemsets (called spatio-sequences) [40], and sequences of spatiotemporal association rules [47].

Cao et al. described the spatiotemporal sequential patterns as the routes of frequently followed by objects in [14]. Namely, a set of frequently visited locations is discovered from a dataset of spatiotemporal trajectory segments. Spatiotemporal sequential patterns are related to the movement patterns of spatiotemporal objects in the form of trajectory segments. Similarly, Giannotti et al. introduced the *trajectory patterns*, and presented an algorithm for mining trajectory patterns [20]. Trajectory patterns represent a set of trajectories frequently visiting similar locations with similar visiting times. While trajectory patterns are concerned with the behavioral aspect of spatiotemporal objects, the term, sequence, refers to visited locations. Verhein introduced complex spatiotemporal sequence pattern mining [47] that focuses on sequences of spatiotemporal association rules. Spatiotemporal association rules represent frequent movements of objects appearing between two regions during a time interval.

Apart from those, Zhang et al. proposed *Splitter* algorithm, which discovers fine-grained sequential patterns from semantic trajectories [51]. The *Splitter* first retrieves spatially coarse patterns, and later reduces them to fine-grained patterns. The discovered patterns are sequences of categorized locations (deduced from semantic trajectories). Another example of spatiotemporal sequences, called spatio-sequences, are presented by Salas et al. [40]. The spatio-sequences are the temporal sequences of ordered spatial itemsets that are used for coupling geographically neighboring phenomena.

Huang et al. presented a framework for mining sequential patterns from spatiotemporal event datasets in [24]. The sequential patterns, in [24], refer to a sequence of event types from point-based event instances. They defined a follow relation between the point-based event instances of two different event types, presented significance measures for sequences, and introduced two pattern-growth based algorithms for the mining task. Both of the algorithms create a pattern tree and expand the nodes of the pattern tree with recursively calling tree expansion procedures (i.e., follow joins). Moreover, Mohan et al. [34] introduced the cascading spatiotemporal pattern mining, which are partially ordered subsets of spatiotemporal event types whose instances are located together and occur in stages.

In [4, 5], Aydin et al. introduced the STES mining from evolving regions. STES mining is also concerned with the sequences of event types, and the instances are trajectories of evolving regions. Hence, the follow relationship is more complex when compared to [24].

The earlier event sequence mining algorithms [4, 5, 24] operate using a set of user-defined thresholds. Here, we will concentrate on discovering the sequences without these threshold values, which often times are not available to domain experts. In STES mining, our focus is mining patterns from evolving region trajectories, and we are not particularly interested in point trajectory data, or stationery spatiotemporal data. Our scope is on mining the most relevant. Our primary objective is to improve the quality of the results and alleviate the issues associated with using preset, and usually arbitrary, thresholds.

1.4 Outline

The rest of this paper is organized as follows. In Section 2, we present background information on STES mining. In Section 3, we introduce our novel STES mining algorithm.

In Section 4, we present our experimental evaluation. Lastly, in Section 5, we present our conclusion and possible future work.

2 Problem formulation

Spatiotemporal event *instances* (ins_i) are the chronologically ordered lists of timestamp-geometry pairs (tg_{ik}). The geometries are region-based and represented with polygons. The instances are evolving region trajectories, and each of them is identified by a unique identifier and has an associated event type. An *event type* signifies the class of its associated instances. A timestamp-geometry pair is a pair of timestamp value (t_i), and a region geometry (g_i). The event type of an instance is represented with $ins_i.\mathcal{E}$.

$$ins_i = \{tg_{i1}, tg_{i2}, \dots, tg_{ik}\} \text{ where } tg_i = \langle t_i, g_i \rangle \quad (1)$$

An event type is denoted by e_j . The set of instances of type e_j is represented as $\mathbb{I}_{e_j}$. In the upper portion of Fig. 2, we show the organizational structure of instances and events. Note here that the set of all instances ($\mathbb{I}$) are essentially union of instances from all event types in the dataset ($\cup \mathbb{I}_{e_j}$ for all e_j).

Let $\mathbb{E} = \{e_1, e_2, \dots, e_k\}$ be the set of all event types, and $\mathbb{I}$ be a database of all event instances. The problem of STES mining is finding frequently occurring sequences of event types (i.e., event sequences) in the form of $(e_{i1} \triangleright e_{i2} \triangleright \dots \triangleright e_{ik})$, such that the instances of e_{i1} is followed-by the instances of e_{i2} , ..., and the instances of e_{ik-1} is followed-by the instances of e_{ik} . The event sequences are denoted as ESq_i , and are derived from instance sequences.

$$\begin{aligned} ES_i &= (e_{i1} \triangleright e_{i2} \triangleright \dots \triangleright e_{ik}) \\ ISq_i &= (ins_{i1} \blacktriangleright ins_{i2} \blacktriangleright \dots \blacktriangleright ins_{ik}) \end{aligned} \quad (2)$$

An instance sequence (denoted as ISq_i in Eq. 2) represents a chain of spatiotemporal *follow* relationships (denoted as “ $\blacktriangleright$ ”) that occur between its participating instances. The number of participating instances in an ISq is the length of the sequence. A length- k instance sequence is alternatively called k -sequence. An ISq_i is *of-type* an event sequence ES_j (as shown in Eq. 3, if and only if the event types of the participating instances of ISq_i are identical and in the same order as the event types in ES_j).

$$ISq_i \text{ is of-type } ES_j \Leftrightarrow ins_{i1}.\mathcal{E} = e_{j1} \wedge \dots \wedge ins_{ik}.\mathcal{E} = e_{jk} \quad (3)$$

In the lower portion of Fig. 2, we schematically depict the follow relation between the instances and show an example of a length-3 instance sequence and its associated spatiotemporal event sequence.

Given this information, the task of STES mining, in general, is interested in discovering spatiotemporal event sequences whose instance sequences are frequently repeated. The instance sequences are discovered by finding significant follow relationships outlined in Section 2.1 and event sequences are derived from the types of these instance sequences. The prevalence of event sequences are measured by participation index measure, which is described in Section 2.2. In this paper, we will focus on mining STESs using a randomization approach, which will take a set of spatiotemporal event instances as input and returns all the discovered STESs together with a list of estimated participation index values for each STES, obtained from randomized trials.

Fig. 2 The hierarchical organization of basic concepts in STES mining

2.1 Follow relationship

The instance sequences are formed by two or more instances. Between each two consecutive instances in an instance sequence, there exists a spatiotemporal *follow* relationship. The simplest form of *follow* relationship occurs between two event instances, and denoted by the ' $\triangleright$ ' symbol. The relationship is characterized by two predicates that are temporal continuity and spatial proximity. To actualize these predicates, we present two concepts that are the *head* and *tail* window of instances.

Head and tail window generation We determine the head and tail with head and tail ratio parameters (denoted as hR and tR). The hR is the ratio between the head segment's lifespan and the instance's lifespan. The tR is the ratio between the tail segment's lifespan and the instance's lifespan.

The head and tail window is derived from the tail of the instance. The first operation for obtaining the tail window is the spatial buffer. Secondly, the spatially buffered geometries are propagated temporally. The amount of buffering is determined by buffer distance parameter (denoted as bd), while the period of temporal propagation is called tail validity period (denoted as tv). Head windows are created similarly, where firstly the head segment

is buffered using the buffer distance parameter and later using the head validity (h_v) parameter the head is propagated. The difference is that the tail window is propagated forward (i.e., towards a later time step) while the head window is propagated backward (i.e., towards an earlier time step). We show an example head and tail segment generation in Fig. 3a and the creation of tail window in Fig. 3b.

Formalization Given two instances ins_i and ins_j , there exists a spatiotemporal *follow* relationship between ins_i and ins_j ($ins_i \triangleright ins_j$) if and only if (1) the start time of ins_i is less than the start time of ins_j , and (2) there exists a spatiotemporal co-occurrence between the tail window of ins_i and the head window of ins_j . Under these conditions, ins_i is the *followee* and ins_j is the *follower* in the relationship.

To form a 2-sequence, there must be one spatiotemporal *follow* relationship between two instances. More generally, to form a k -sequence, there must be $k - 1$ spatiotemporal *follow* relationships between each consecutive participating instance.

2.2 Significance measurement

To measure how frequent an STES is in a given dataset, we use the participation index. The participation index is defined in [23], and shows the minimum relative frequency of the participating event types. For an event sequence, $ES_j = (e_{j_1} \triangleright \dots \triangleright e_{j_k})$, the participation index of an STES (ES_j) is the minimum of the participation ratios (pr) of its event types.

$$pi(ES_j) = \min(pr(e_{i_1}|ES_j), \dots, pr(e_{i_k}|ES_j)) \quad (4)$$

The participation ratio of an event type (e_i) on an STES (ES_j) is the ratio of the number of unique participators of e_i 's instances to the total number of event instances of e_i in the dataset.

$$pr(e_i|ES_j) = \frac{|\{ins_i | ins_i \in ISq_i \wedge ins_i.\mathcal{E} = e_i \wedge ISq_i \text{ of-type } ES_j\}|}{|\mathbb{I}_{e_i}|} \quad (5)$$

where $|\cdot|$ shows the set size.

Fig. 3 **a** Creating the head and tail segments. **b** Creating the tail window of an instance from its tail using buffering and forward temporal propagation

Another aspect of significance is the strength of the follow relationships. The significance assessment is important as the accuracy and reliability of resulting event sequences are dependent on the discovered significance of the instance sequences. For assessing the strength of the *follow* relationships, we use the *chain index* (denoted as *ci*). The *ci* for 2-sequences is defined as the significance of spatiotemporal co-occurrence between the tail window (*tw*) of followee and the head window (*hw*) of the follower. In this work, we measure the significance of spatiotemporal co-occurrence using J^* measure [6, 8].

Formally, for a 2-sequence, $ISq_r = (ins_{r1} \blacktriangleright ins_{r2})$, the strength of the *follow* relationship is assessed as:

$$ci(ISq_r) = \begin{cases} J^*(tw_{r1}, hw_{r2}) & \text{if } ins_{r1}.t_s < ins_{r2}.t_s, \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

where t_s represents the starting time of an instance

For a *k*-sequence $ISq_i = (ins_{i1} \blacktriangleright ins_{i2} \blacktriangleright \dots \blacktriangleright ins_{ik})$, where $k > 2$, the significance is assessed as the minimum chain index of each follow relationship in the instance sequence (Eq. 7)

$$ci(ISq_i) = \min_{1 \leq j < k} (ci(ins_{ij} \blacktriangleright ins_{i(j+1)})) \quad (7)$$

In the threshold-based approaches, the instance sequences are considered significant if their chain index (*ci*) value is greater than a user-defined chain index threshold (ci_{th}). Similarly, the event sequences are considered frequent if their participation index (*pi*) value is greater than a user-defined participation index threshold (pi_{th}).

2.3 Threshold-based STES mining approach

The pattern-growth based ESGROWTH algorithm is introduced in [4]. The algorithm firstly identifies the *follow* relationships and creates the event sequence graph (*ESG*). Later, the algorithm recursively discovers the STESs from the graph structure. The ESGROWTH algorithm is outlined in Algorithm 1.

In the initialization steps, as explained in Section 2, the algorithm creates heads and tail windows of instances and identifies the follow relationships between those instances (Algorithm 1–Lines 2 to 4). The follow relationships are discovered by a spatiotemporal join procedure where the head and tail window trajectories of the instances are joined and filtered based on the ci_{th} for significance testing. Later, these significant follow relationships are inserted into a directed acyclic graph structure – *ESG* (Line 5). For any discovered follow relationship, $(ins_i \blacktriangleright ins_j)$, the TRANSFORM procedure adds two vertices that represent ins_i and ins_j (if they are not present already), and a directed edge from ins_i to ins_j . The graph only stores the identifiers (*i* and *j*) and event types ($ins_i.\mathcal{E}$ and $ins_j.\mathcal{E}$) of the instances in its vertices.

After creating *ESG*, the ESGROWTH recursively discovers the STESs. The algorithm starts by iterating the event types (e_i) in the set of all events ($\mathbb{E}$). For each event type (e_i), it identifies the non-leaf vertices, which corresponds to the instances of e_i (Step 4). Then a participation ratio (*pr*) is calculated from those vertices to check the prevalence using pi_{th} . If *pr* is greater than the pi_{th} , the recursive GROWSEQUENCE procedure is called.

The GROWSEQUENCE procedure is shown in the second part of Algorithm 1. The procedure takes a prefix event sequence (*prefix*), the current minimum *pr* for the prefix sequence, and a set of pointers to the vertices (V_{pre}) as parameters. The vertices in V_{pre} correspond to the last discovered vertices in the paths, which virtually forms the instance sequences of-type *prefix* (event sequence) in the *ESG*. The procedure proceeds as follows. First, the

successors (immediate neighbors) of the vertices in V_{pre} are found and added to the successor vertex set ($SucV$). Then, for each event type e_j , a subset of successor vertex set (containing the instances of e_j) is created (Line 4 of GROWSEQUENCE procedure in Algorithm 1—denoted as $SucV_{e_j}$). After identifying successors, a temporary participation ratio value (pr') is calculated for extended event sequence (Line 5 of GROWSEQUENCE). If pr' is greater than the pi_{th} , the prefix is extended with e_j , and a new prefix event sequence (denoted as $prefix'$) is created (Line 7 of GROWSEQUENCE procedure).

Algorithm 1 Finding the prevalent STESS from the event sequence graph (ESGROWTH).

Input: Set of event instances – $\mathbb{I}$, Chain index threshold – ci_{th} , Participation index threshold – pi_{th}

Variables: Set of event types – $\mathbb{E}$, Current participation ratio – pr , Non-leaf vertices of type e_i in $ESG - V_{e_i}$

Output: Set of prevalent STESS – ES .

```

1: procedure ESGROWTH( $\mathbb{I}$ ,  $ci_{th}$ ,  $pi_{th}$ )
2:    $H \leftarrow \text{CREATEHEADS}(\mathbb{I})$ 
3:    $TW \leftarrow \text{CREATETAILWINDOWS}(\mathbb{I})$ 
4:    $followRelationships \leftarrow \text{STJOIN}(H, TW, ci_{th})$ 
5:    $ESG \leftarrow \text{TRANSFORM}(followRelationships)$ 
6:    $ES \leftarrow \{\}$ 
7:   for all  $e_i \in \mathbb{E}$  do
8:      $V_{e_i} \leftarrow \text{FINDINSTANCESOF}(e_i, ESG)$ 
9:      $pr \leftarrow \text{FINDPARTICIPATIONRATIO}(e_i, V_{e_i})$ 
10:    if  $pr > pi_{th}$  then
11:       $\text{GROWSEQUENCE}(e_i, pr, V_{e_i})$ 
12:    end if
13:  end for
14:  return  $ES$ 
15: end procedure

```

```

1: procedure GROWSEQUENCE( $prefix$ ,  $pr$ ,  $V_{pre}$ )
2:    $SucV \leftarrow \text{FINDSUCCESSORSOF}(V_{pre}) // \text{in } ESG$ 
3:   for all  $e_j \in \mathbb{E}$  do
4:      $SucV_{e_j} \leftarrow \text{FINDINSTANCESOF}(e_j, SucV)$ 
5:      $pr' \leftarrow \text{FINDPARTICIPATIONRATIO}(e_j, SucV_{e_j})$ 
6:     if  $pr' > pi_{th}$  then
7:        $prefix' \leftarrow prefix \triangleright e_j$ 
8:        $ES.\text{INSERT}(prefix', pr)$ 
9:        $\text{GROWSEQUENCE}(prefix', \min(pr, pr'), SucV_{e_j})$ 
10:    end if
11:  end for
12: end procedure

```

At this point, $prefix'$ is guaranteed to be prevalent, thus is inserted to the set of prevalent event sequences, ES . Lastly, the GROWSEQUENCE procedure is called with the newly created event sequence, $prefix'$. Along with the $prefix'$, the minimum of old and new participation ratio ($\min(pr, pr')$), and the vertex subset formed by the successors ($SucV_{e_j}$) are passed as parameters. Note that the base case in the recursion occurs when the temporary

participation ratio is less than the participation index threshold. In this case, the *prefix*', which is created by appending the new event type to *prefix*, is not prevalent. Therefore, there is no need to check the longer length event sequences generated from *prefix*'.

3 A novel randomization approach

Previous approaches [4, 5, 9] use threshold-based STES mining algorithms that heavily relies on the domain experts' knowledge to choose some appropriate threshold parameters, which is usually not available. The thresholds in the earlier approaches are necessary to understand whether a discovered sequence is spurious or sufficiently frequent. However, often times, determining generic threshold values for all the event sequences is impractical, even for the domain experts. To tackle these issues, we propose a novel algorithm, RAND-ESMINER (**randomized spatiotemporal event sequence miner**) that can help domain experts understand the intrinsic characteristics of the spatiotemporal event sequences better.

The threshold-based approach is essentially a constraint-based data mining approach, where the instance sequences are filtered based on the ci_{th} , and the event sequences are filtered based on the pi_{th} . The usefulness of these thresholds for the efficiency of the mining algorithms is indisputable, primarily because of the exponential time and space complexity of these algorithms. However, the imbalance of the spatiotemporal instances in these datasets and their characteristics such as lifespan, total area, or the areal evolution make it very difficult for domain or data mining experts to create meaningful threshold values [7]. Because of these, we created an algorithm that does not take participation index or chain index thresholds as input, but outputs STESs along with a distribution of pi values. Threshold-based algorithms output a set of prevalent STESs (often coupled with a single pi value). From a practical point of view, event sequences which include rarely occurring events or instances with significantly different spatiotemporal characteristics are often overlooked. With the randomization approach, we do not consider any significance or prevalence threshold and perform the mining on the resampling subsets of the follow relationships. The randomized algorithm is inspired from the permutation tests (random resamples without replacement) and outputs the participation index values of all the discovered patterns.

3.1 Permutation tests

The primary task of the statistical descriptors is to summarize a characteristic of the given data and generalize the finding to the larger population. Basic sample statistics such as sample mean or median give information about that particular sample; however, their values would fluctuate from sample to sample and magnitude of these fluctuations around the corresponding parameter is also important for the relevance of these results.

Statistical bootstrapping is an alternative to the traditional statistical methodology of assuming a particular probability distribution for a sample. Bootstrap is a random resampling technique that estimates the distribution of a statistic and allows measures of accuracy to sample estimates [17]. It is especially useful when there is no analytical form to help estimate the distribution of the statistics of interest such as mean or variance.

An increasingly common statistical tool for constructing sampling distributions is the randomization test (also referred to as permutation test). Similar to the idea of bootstrapping, a permutation test builds sampling distribution, which is the permutation distribution, by resampling the observed data. Specifically, we can permute the observed data. Unlike

bootstrapping, permutation tests resamples the data without replacement, which is more appropriate for our tasks.

In our application area of solar data mining, the lack of accurate data is a common problem and one way to tackle with these noisy data is to randomize the mining process and obtain uncertainties with confidence intervals. Our primary data source for STES mining is the solar event metadata obtained from the feature finding teams (FFT) of Solar Dynamics Observatory [33]. Firstly, most of the solar event instances (representing the regions of solar events) have become available only after the launch of the Solar Dynamics Observatory (SDO). Hence, we only have the slices of the data. Additionally, the SDO only captures the images of one side of the sun that is visible to us, and we currently do not have the reports of the solar events occurring on the opposite side of our line of sight.

In our randomization approach, we treat the follow relationships as the sample dataset, and the participation index (pi) values of STESs as a complex statistic to be obtained from the *ESG* structure. By applying a random resampling without replacement (i.e., applying a set of permutation test) of the follow relationships (i.e., edges) we have the opportunity to explain the prevalence of STESs as a distribution. Note here that we do not perform a traditional permutation test which would require a null model and a statistical hypothesis testing. Given the characteristics of solar event datasets, this approach is very promising due to its power of estimating the confidence intervals of the participation index for STESs. This is to say, for each randomized experiment, we can obtain a participation index value for a discovered pattern and estimate its likelihood and see the variance of these values.

3.2 Randomized algorithm: RAND-ESMINER

In this part, we will explain the details of RAND-ESMINER algorithm. The algorithm makes use of random resampling of follow relationships and outputs all the discovered STESs along with their participation index values, which shows the prevalence of a particular STES, in each random trial.

In Algorithm 2, we show the overview of the Randomized STES mining algorithm. The algorithm initially discovers all the follow relationships as in the threshold-based ESGROWTH algorithm and creates the *ESG* (Lines 2 to 5). In a nutshell, it firstly randomly resamples the edges in the *ESG*, then discovers the STESs from the resampled event sequence graphs, and eventually collects the results. The algorithm takes the set of instances ($\mathbb{I}$), resampling ratio (rR), and the number of random trials to be performed (ν) as input. Resampling ratio (rR) is the ratio between the number of edges to be resampled and total number of edges in the *ESG*. Note here that neither rR nor ν parameters require the expert knowledge; they are used for regulating the randomized trials and are not necessarily dependent on the intrinsic characteristics of data.

The *ESG* structure allows us to create random resamples of the *follow* relationships. The RAND-ESMINER algorithm performs a randomized run for ν times to estimate the pi value for all the discovered STESs (Lines 7 to 15). The resampling is applied on the edges of the graph, which corresponds to the follow relationships among instances. Edge resampling creates a subgraph of the *ESG*, that is $rESG$, by randomly sampling from the edge set of *ESG* without replacement (Line 8). Note that our graph structure is not a multi-graph; therefore, we opt for a permutation test (resampling without replacement) instead of bootstrapping (resampling with replacement).

Next, we discover the STESs from the resampled subgraphs of *ESG*. For each resampled subgraph, we perform a recursive procedure similar to the ESGROWTH algorithm.

For each event type e_i , we find the non-leaf vertices of e_i , and grow the sequences (See GROWRANDSEQUENCES procedure in the second part of Algorithm 2). This can be considered as running the ESGROWTH algorithm with $pi_{th} = 0.0$. Lastly, we append them to the map structure (ES_{rand}) for every iteration, and return the ES_{rand} , which contains the discovered STESs and a size- (v) list of pi values for each sequence. If an STES is not discovered during a trial, the pi value for that STES is recorded as 0. If an STES is recorded for the first time during the k th trial, we create a new list of pi values (length- k) backfilled with zeroes for each previous trial it was not discovered.

Algorithm 2 RAND-ESMINER algorithm.

Input: Set of event instances – $\mathbb{I}$, resample ratio – rR , number of randomization trials – v

Variables: Resampled subgraph of the ESG – $rESG$, intermediate event sequences – iES

Output: STESs and pi 's – $ES_{rand} = \langle ES, (pi_j, \dots) \rangle$.

```

1: procedure RAND-ESMINER( $\mathbb{I}, rR, v$ )
2:    $H \leftarrow \text{CREATEHEADS}(\mathbb{I})$ 
3:    $TW \leftarrow \text{CREATETAILWINDOWS}(\mathbb{I})$ 
4:    $followRelationships \leftarrow \text{STJOIN}(H, TW, ci_{th})$ 
5:    $ESG \leftarrow \text{TRANSFORM}(followRelationships)$ 
6:    $ES_{rand} \leftarrow \{\}$ 
7:   for  $j$  from 1 to  $v$  do
8:      $rESG \leftarrow \text{EDGERESAMPLE}(ESG, rR)$ 
9:      $iES \leftarrow \{\}$ 
10:    for all  $e_i \in \mathbb{E}$  do
11:       $V_{e_i} \leftarrow \text{FINDINSTANCESOF}(e_i, rESG)$ 
12:      if  $V_{e_i} \neq \emptyset$  then
13:         $pr \leftarrow \text{FINDPARTICIPATIONRATIO}(e_i, V_{e_i})$ 
14:         $\text{GROWRANDSEQUENCE}(e_i, pr, V_{e_i}, iES)$ 
15:      end if
16:    end for
17:     $ES_{rand}.\text{APPEND}(iES)$ 
18:  end for
19:  return  $ES_{rand}$ 
20: end procedure

```

```

1: procedure GROWRANDSEQUENCE( $prefix, pr, V_{pre}, iES$ )
2:    $SucV \leftarrow \text{FINDSUCCESSORSOF}(V_{pre}) // \text{in } ESG$ 
3:   for all  $e_j \in \mathbb{E}$  do
4:      $SucV_{e_j} \leftarrow \text{FINDINSTANCESOF}(e_j, SucV)$ 
5:     if  $SucV_{e_j} \neq \emptyset$  then
6:        $pr' \leftarrow \text{FINDPARTICIPATIONRATIO}(e_j, SucV_{e_j})$ 
7:        $prefix' \leftarrow prefix \triangleright e_j$ 
8:        $iES.\text{INSERT}(prefix', pr)$ 
9:        $\text{GROWRANDSEQUENCE}(prefix', \min(pr, pr'), SucV_{e_j}, iES)$ 
10:    end if
11:  end for
12: end procedure

```

It is worth mentioning that unlike the threshold-based approaches which return a list of prevalent STESSs, the output of the RAND-ESMINER algorithm is a map structure whose keys are discovered patterns and the values are list of calculated pi values in each of the (v) random trials.

4 Experimental evaluation

4.1 Experimental settings

In this section, we present our experimental evaluations of the randomization-based STES mining approach. We used real-life datasets from solar astronomy field. We evaluated the runtime performance of graph transformation procedures and compared the ESGROWTH and RAND-ESMINER algorithms.

Our algorithms are implemented in Java programming language, datasets were stored in text files, and experiments were conducted on an Ubuntu virtual machine with 1TB RAM with an Intel Xeon processor. All the event instances and graph elements are stored in memory for fair comparison.

4.2 Datasets and configurations

To analyze the performance of our proposed algorithm, we used real-life solar event datasets. These are monthly datasets from 2012, which include the spatiotemporal instances of seven different solar event types that are: Active Regions (ar), Coronal Holes (ch), Emerging Flux (ef), Filaments (fi), Flares (fl), Sigmoids (sg), and Sunspots (ss). Each instance consists of region polygons, downloaded from Heliophysics Event Knowledgebase (HEK) [31], and the regions are tracked and interpolated using the algorithms presented in [11, 26]. The characteristics of our real-life datasets can be seen in Table 1. The datasets are in tabular format, where the instances of a particular event type is stored as a table. Each row shows a particular time-geometry pair with four attributes: instance identifier, start time of the time-geometry pair, end time of the time-geometry pair, and spatial geometry. The spatial geometry is a polygon object formatted as well-known text (WKT) format.

The goal of the experiments is to examine the performance of our randomized algorithm, both in terms of relevance and efficiency, on these datasets and compare it with the threshold-based ESGROWTH algorithm. We will compare the discovered STESSs on our relevance analysis and later evaluate the running time efficiency. A preliminary step of the algorithm is the initialization (head and tail window generation) and the graph transformation. For that reason, we kept the global parameters that are used in the ESG creation as constant throughout all the experiments. These parameters represent independent variables that should not alter the performance of the algorithms within these set of experiments. Head and tail ratio parameters were selected as 0.2 and 0.5, respectively. The value used for the tail validity (tv) is 4 hours. Head validity (hv) was set to zero for consistency with our earlier works. We chose 25 $arcsec$ as the buffer distance (bd) parameter.

For the case of the threshold-based algorithm, we conducted 16 experiments for each dataset with varying ci_{th} nad pi_{th} values. The ci_{th} value was set to 0.10, 0.15, 0.2, and 0.25, while pi_{th} value was set to 0.04, 0.08, 0.12, and 0.16. We ran the ESGROWTH algorithm with the above-mentioned combinations of ci_{th} and pi_{th} values to discover the frequent

sequences. Eventually, a total of 192 experiments were performed on the 12 datasets for the threshold-based approach. On the other hand, for the randomization approach, we resampled the data 100 times ($\nu = 100$) for every dataset and estimated the distribution of the pi values of all the event sequences. The size of each sample is 10% the size of its respective original dataset ($rR = 0.1$). Thus, we generated 100 pi values for all the discovered sequences to estimate prevalence of event sequences.

4.3 Relevance analysis

In this part, we will discuss the relevance of the mining results from RAND-ESMINER algorithm. For brevity, we chose to illustrate the length-2, length-3, and length-4 STESs with the top-15 mean pi values from Jan, Feb, Mar, and Apr datasets. The comprehensive results for length-2, 3, and 4 STESs from all datasets can be found in the [Appendix](#) of this paper.

Figure 4 illustrates the most prevalent length-2 (top row), length-3 (middle row), length-4 (bottom row) STESs from four datasets. The distribution of pi values from RAND-ESMINER are demonstrated with the box plots. We also demonstrate the discovered pi values from the threshold-based approach with varying size scatter elements. Each scatter represents a different experimental run, and when the event sequence is not found to be prevalent on an ESGROWTH experiment (meaning pi was less than pi_{th}), the result from that experiment is omitted.

From Fig. 4, we can see that the discovered top-15 STESs are consistent throughout the four datasets. We also present the number of top-15 occurrences for length-2 STESs from all datasets in Fig. 5. The results from the length-3 and length-4 STESs is available in the [Appendix](#) for completeness. Eight of the top-15 length-2 STESs are discovered in all twelve datasets and 13 of them were discovered at least ten times. We further analyzed these 13 STESs. Table 2 shows the number of times STESs was discovered by ESGROWTH as well as the averaged (across 12 monthly datasets) median pi values and averaged percentage STES was discovered in randomized trials. We can observe that averaged median pi values for these STESs are generally over 0.04 (pi_{th} for the comparable ESGROWTH

Table 1 Characteristics of the solar event datasets

Dataset	Tag	#of Instances	#of Region Polygons
January 2012	Jan	2,082	159,773
February 2012	Feb	1,252	111,615
March 2012	Mar	2,022	157,374
April 2012	Apr	1,784	124,611
May 2012	May	2,255	199,390
June 2012	Jun	2,245	206,442
July 2012	Jul	2,387	182,601
August 2012	Aug	2,046	193,028
September 2012	Sep	2,119	186,906
October 2012	Oct	1,945	178,642
November 2012	Nov	2,057	161,930
December 2012	Dec	1,682	156,333

Fig. 4 Mixed plots of pi values from RAND-ESMINER and ESGROWTH algorithms. Distributions of pi values from RAND-ESMINER are shown as boxplots, while pi values (discovered using ESGROWTH) with different thresholds are shown as circles. Legend on top shows the thresholds for ESGROWTH ci and pi thresholds

Fig. 5 Number of top-15 occurrences for length-2 STESSs, discovered using RAND-ESMINER from twelve monthly datasets

experiment) and they are discovered in almost all of the randomized trials (that is to say the average percentage of randomized trials these STESSs were discovered is above 97% for the aforementioned STESSs). This is not the case for threshold-based runs, where some of these well-known STESSs were not discovered, even for relatively low threshold values. (See an incomplete list of observations found in the literature for some of these patterns: ‘ss > sg’ [12, 45]; ‘ef > ef’ [41]; ‘ar > ar’ [21, 46]; ‘ef > fl’ [2, 41, 48]; or ‘ss > ss’ [22, 43]).

Another aspect of our evaluation is the relevance comparison with threshold-based approach in terms of varying frequencies of STESSs. One observation we can make is the variation of the pi values when using different ci_{th} values in the threshold-based approach. The variation is two-fold: (1) The variation of the pi values for a particular STESS and (2) the variation of the pi values across different STESSs. The latter is much expected as the natural phenomena may or may not be spatiotemporally following each other. However, the former variation poses a challenge that is difficult to solve with trial and error. For example, for ($fl > fl$) sequences $ci_{th} = 0.2$ and $pi_{th} = 0.12$ can be an accurate cut-off points for thresholds. However, if we set the ci_{th} to 0.2 and pi_{th} to 0.12 for the entire dataset, we miss practically all ($ar > fl$) sequences, as well as the sequences including the ($ar > fl$) subsequence. It is well-known to solar physics experts that flares can occur anywhere on the Sun’s surface, from active regions (ar) to the boundaries of the magnetic network of the quiet Sun [27]. However, large area flares (fl) have preferred locations. They originate from the large active regions showing a complex geometry of the 3D magnetic field [38].

Table 2 Comparisons of thirteen length-2 STESs for RAND-ESMINER and ESGROWTH across 12 monthly datasets

Selected	RAND-ESMINER		ESGROWTH $ci_{th} = 0.1/pi_{th} = 0.04$
Length-2 STES	Average Median pi	Average perc. STES discovered	Number of monthly datasets STES discovered
$fl \triangleright fl$	0.297	100.00	12
$ch \triangleright ch$	0.161	99.92	11
$ar \triangleright fl$	0.082	100.00	12
$ss \triangleright ss$	0.129	97.42	10
$sg \triangleright sg$	0.086	100.00	12
$ar \triangleright sg$	0.075	100.00	12
$sg \triangleright fl$	0.062	100.00	11
$ar \triangleright ar$	0.042	99.58	4
$ss \triangleright fl$	0.060	100.00	11
$ef \triangleright fl$	0.048	100.00	9
$ef \triangleright ef$	0.029	100.00	2
$fl \triangleright sg$	0.040	100.00	11
$ss \triangleright sg$	0.039	98.50	2

The STESs are selected based on their top-15 occurrences (See Fig. 5). Averaged median participation index (pi) values and average percentage an STES was discovered in randomized trials are reported for RAND-ESMINER. Number of months an STES was discovered by ESGROWTH (with $ci_{th} = 0.1$ and $pi_{th} = 0.04$) is also reported. RAND-ESMINER discovered these STESs for all monthly datasets

To capture ($ar \triangleright fl$) event sequence, we can use $ci_{th} = 0.1$ and $pi_{th} = 0.04$ (see the results in Table 2). However, this time majority of ($ar \triangleright ar$), ($ef \triangleright ef$), and ($ss \triangleright sg$) would be missed. These examples can be extended to include those three sequences leading to a never-ending cycle of pattern importance discussion. Even for the simplistic cases of ($fl \triangleright fl$) and ($ar \triangleright fl$), or ($ar \triangleright fl$) and ($ss \triangleright sg$) creating user-defined thresholds is difficult, primarily because of the unbalanced spatiotemporal characteristics of the natural phenomena. Therefore, we can suggest that mining a distribution of pi values using random edge resampling from the sample ESG is a better approach, because outputting a mere pi value based on set thresholds cannot properly represent the characteristics of the population.

4.4 Running time analysis for initialization and ESG transformation steps

In this part of our evaluation, we will show the runtime complexity of the initialization phase of RAND-ESMINER algorithm. In Fig. 6, we demonstrate the running times in the initialization phase of the RAND-ESMINER algorithm for all the datasets.

We measured the running times in the initialization phase into two categories: (1) Head and Tail Window Generation Time, which is denoted as *HT Generation Time* in Fig. 6 and corresponds to Line 2 and 3 in Algorithm 2 and (2) Follow Relation and Graph Transformation Time, which is denoted as *Follow Time* in Fig. 6 and corresponds to Line 4 and 5

Fig. 6 Running times (values on major Y-axis) required for head and tail window generation (green line) and identification of the follow relationships (blue line). Bars show the number of edges and vertices in the *ESG* showing the complexity of the *ESGs* (values on to secondary Y-axis)

in Algorithm 2. The head and tail window generation requires complex spatial buffer and union operations. Similarly, the follow relationships are discovered with a computationally expensive spatiotemporal join operation on evolving region trajectories. Creating the *ESG* is significantly less complex in terms of computation time.

Along with the running times, in Fig. 6, we illustrated the vertex and edge counts in the created *ESG* for each dataset. The number of vertices correspond to the number of instances in the dataset, while the number of edges show the number of follow relationships.

From the results, we can see that the head and tail window generation time varies significantly for each monthly dataset. We can observe that part of this stems from the number of instances (vertices) in the dataset, and another factor is the number of individual region polygons in the datasets. We observe the highest head and tail window generation times are recorded for May and June datasets where we have highest number of region polygons in the datasets. Similarly, the lowest HT generation times are recorded for February and April datasets where we have the lowest number of region polygons.

The follow time also greatly varies across our datasets. The follow time depends on the number of spatiotemporal follow relationships among the instances in the dataset. While we cannot assert a total correlation, the number of edges in the generated graph is a good indicator for the follow time. Another factor that impacts the follow time is the number of instances and region polygons, because we get 20% and 50% of the instance trajectories as heads and tails (as $hR = 0.2$ and $tR = 0.5$).

For the case of the head and tail-window generation, our algorithm iterates through all the instances in the dataset and compute the time propagated and spatially buffered time-geometry pairs (representing the region trajectories). This process is done in linear time which explains the relation between the running time and the number of instances and region polygons. On the other hand, the *ESG* generation algorithm iterates through the tail windows and performs a spatiotemporal join on overlap predicate with the head window

of instances. This makes the complexity of the follow relationship identification quadratic; however, since we apply a two-step filter based on time-overlap and spatial-overlap predicates the complexity becomes subquadratic (and very close to linear) with respect to the number of region polygons in follow time. It should be noted that, in the situation where there is a time requirement constraint, the user can shrink the size of the head and tail windows to decrease the amount of overlapping; thus, reducing the number of follow relationships.

4.5 Running time analysis of RAND-ESMINER

In this part of our experiments, the running time requirements of our RAND-ESMINER algorithm will be compared to the ESGROWTH algorithm. In Fig. 7, we demonstrate the total running times of our algorithms for each dataset (in (a)), as well as the average time spent on mining STESs from *ESG* (in (b)). In Fig. 7a, the blue bars show the average running time of ESGROWTH algorithm with 19 different ci_{th} values. The red bars show the total running time of RAND-ESMINER algorithm, which consists of 100 randomized runs on the *ESG*. In Fig. 7b, we demonstrate the running times required for mining STESs from *ESG*. The initialization steps (HT generation and follow times shown in Fig. 6) are omitted here for a better comparison, and we report the average running times of threshold-based runs (with 16 different ci_{th} and pi_{th} combinations), and the average running time of 100 randomized runs for each dataset.

From the results shown in Fig. 7a, we can notice that the total running times required for mining the STESs follow a very similar pattern to the initialization, and it can be observed that for threshold-based approach, the total running time is dominated by the initialization. The higher ci_{th} values we use for filtering the insignificant follow relationships (or edges) extensively prune the *ESG*, leading to very low graph mining times. Nevertheless, it is difficult to make conclusions about the trustworthiness of the STESs or overall generality of STES mining process with high ci_{th} values. When we analyze the performance of the RAND-ESMINER algorithm, we see that for sparse *ESGs* (such as Feb, Apr, and Oct datasets) the total running time of the RAND-ESMINER is similar to the ESGROWTH. On the other hand, for denser *ESGs* (such as May, Jun, and Jul datasets), we observe greater

Fig. 7 a Total running time of RAND-ESMINER vs. average total running time ESGROWTH algorithm b Comparison of the average run on the randomized and threshold-based approaches

differences. This can be explained well with the algorithmic setup of randomization approach and the observations from Fig. 7b. The average *ESG* mining time of randomization approach for May, Jun, and Jul datasets are relatively higher than other datasets. In our experiments, the *ESG* is resampled 100 times, and total running time of RAND-ESMINER includes all the 100 randomized runs. Whereas, for the threshold-based approach, the *ESG* is mined only once.

In summary, the total running time required for RAND-ESMINER is approximately 27% more than ESGROWTH. The running time required for the RAND-ESMINER is primarily dependent on the resampling ratio and the number of trials. To increase the trustworthiness of the results, one can increase the number of trials and resampling ratio. In addition to that, the trustworthiness of the results can be traded off with the running time. Choosing a lower resampling ratio or number of trials would decrease the running time, as well as the trustworthiness of the results.

5 Conclusion and future work

In this work, we have introduced a novel spatiotemporal event sequence mining algorithm – RAND-ESMINER, specifically designed for discovering STESs without user-defined thresholds. Our method differs from the conventional threshold-based methods that can be inaccurate; thus, inapplicable for large-scale data analysis. Our novel randomized algorithm relies on applying permutation tests to the edges in the event sequence graph generated by spatiotemporal follow relationships. Unlike the traditional techniques which discover STESs with one *pi* value [4, 5], our algorithm discovers a distribution of *pi* values, and estimates a confidence interval for STESs without any thresholds. Mining STESs without thresholds is significant for scientific fields, as it can be easily applied for explorative tasks.

Our future work lies in the parallelization of RAND-ESMINER algorithm. As the number of random resamplings and resampling ratio increases, the RAND-ESMINER can be less efficient and exploiting parallel computation can leverage the efficiency issues and provide us with highly robust outcomes.

Acknowledgements This project has been supported in part by funding from the Division of Advanced Cyberinfrastructure within the Directorate for Computer and Information Science and Engineering, the Division of Astronomical Sciences within the Directorate for Mathematical and Physical Sciences, and the Division of Atmospheric and Geospace Sciences within the Directorate for Geosciences, under NSF award #1443061. It was also supported in part by funding from the Heliophysics Living With a Star Science Program, under NASA award #NNX15AF39G.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix: Comparisons of *pi* values for RAND-ESMINER and ESGROWTH

A.1 Size-2 STESs

Fig. 8 Mixed plots of pi values from RAND-ESMINER and ESGROWTH algorithms for top-15 length-2 STES from all the datasets. Distributions of pi values from RAND-ESMINER are shown as boxplots, while pi values (discovered using ESGROWTH) with different thresholds are shown

A.2 Size-3 STESS

Fig. 9 Mixed plots of π values from RAND-ESMINER and ESGROWTH algorithms for top-15 length-3 STESS from all the datasets. Distributions of π values from RAND-ESMINER are shown as boxplots, while π values (discovered using ESGROWTH) with different thresholds are shown

A.3 Size-4 STEs

Fig. 10 Mixed plots of pi values from RAND-ESMINER and ESGROWTH algorithms for top-15 length-4 STEs from all the datasets. Distributions of pi values from RAND-ESMINER are shown as boxplots, while pi values (discovered using ESGROWTH) with different thresholds are shown

A.4 Number of top-15 occurrences for length-3 and length-4 STESS

Fig. 11 Number of top-15 occurrences for length-3 STESSs, discovered using RAND-ESMINER from twelve monthly datasets

Fig. 12 Number of top-15 occurrences for length-4 STESSs, discovered using RAND-ESMINER from twelve monthly datasets

References

- Ahmed N, Michelin RA, Xue W, Ruj S, Malaney R, Kanhere SS, Seneviratne A, Hu W, Janicke H, Jha SK (2020) A survey of COVID-19 contact tracing apps. *IEEE Access* 8:134577–134601. <https://doi.org/10.1109/access.2020.3010226>
- Archontis V (2012) Magnetic flux emergence and associated dynamic phenomena in the sun. *Philos Trans R Soc A: Math Phys Eng Sci* 370(1970):3088–3113
- Aydin B, Angryk R (2015) Spatiotemporal frequent pattern mining on solar data: current algorithms and future directions. In: 2015 IEEE international conference on data mining workshop (ICDMW). IEEE. <https://doi.org/10.1109/icdmw.2015.10>
- Aydin B, Angryk R (2016) A graph-based approach to spatiotemporal event sequence mining. In: IEEE international conference on data mining workshops, ICDM workshops 2016, Dec. 12–15, 2016, Barcelona, Spain, pp 1090–1097
- Aydin B, Angryk R (2016) Spatiotemporal event sequence mining from evolving regions. In: 23rd International conference on pattern recognition, ICPR 2016, Cancun, Mexico, December 4–8, 2016
- Aydin B, Akkineni V, Angryk R (2015) Time-efficient significance measure for discovering spatiotemporal co-occurrences from data with unbalanced characteristics. In: Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems, Bellevue, WA, USA, Nov. 3–6, 2015, pp 80:1–80:4
- Aydin B, Angryk R, Filali Boubrahimi S, Hamdi S (2016) Spatiotemporal frequent pattern discovery from solar event metadata. In: AGU Fall meeting abstracts
- Aydin B, Kucuk A, Angryk RA, Martens PC (2017) Measuring the significance of spatiotemporal co-occurrences. *ACM Trans Spat Algorithms Syst (TSAS)* 3(3):1–35

9. Aydin B, Kucuk A, Boubrahimi SF, Angryk RA (2017) Top-(r%, K) spatiotemporal event sequence mining. In: 2017 IEEE international conference on data mining workshops, ICDM Workshops 2017, New Orleans, LA, USA, November 18–21, 2017, pp 250–257. <https://doi.org/10.1109/ICDMW.2017.39>
10. Baker DN, Balstad R, Bodeau J, Cameron E, Fennel J, Fisher G, Forbes K, Kintner P, Leffler L, Lewis W et al (2008) Severe space weather events-understanding societal and economic impacts. a workshop report. Tech. rep. The National Academies Press
11. Boubrahimi SF, Aydin B, Kempton D, Angryk RA (2016) Spatio-temporal interpolation methods for solar events metadata. In: 2016 IEEE international conference on big data, BigData 2016, Washington DC, USA, December 5–8, 2016, pp 3149–3157
12. Brown D, Nightingale R, Alexander D, Schrijver C, Metcalf T, Shine R, Wolfson C et al (2003) Observations of rotating sunspots from trace. *Solar Phys* 216(1–2):79–108
13. Campisano R, Borges H, Porto F, Perosi F, Pacitti E, Massegia F, Ogasawara E (2018) Discovering tight space-time sequences. In: International conference on big data analytics and knowledge discovery. Springer, pp 247–257
14. Cao H, Mamoulis N, Cheung DW (2005) Mining frequent spatio-temporal sequential patterns. In: Proceedings of the 5th IEEE international conference on data mining (ICDM 2005), 27–30 November 2005, Houston, Texas, USA, pp 82–89
15. Delafontaine M, Versichele M, Neutens T, Van de Weghe N (2012) Analysing spatiotemporal sequences in bluetooth tracking data. *Appl Geogr* 34:659–668
16. Dong G, Pei J (2007) Sequence data mining, vol 33, Kluwer. <https://doi.org/10.1007/978-0-387-69937-0>
17. Efron B (1981) Nonparametric estimates of standard error: the jackknife, the bootstrap and other methods. *Biometrika* 68(3):589–599
18. Erwig M (2004) Toward spatio-temporal patterns. In: de Caluwe R, de Tré G, Bordogna G (eds) Spatio-temporal databases. Springer, Berlin, pp 29–53. https://doi.org/10.1007/978-3-662-09968-1_3
19. Ferretti L, Wymant C, Kendall M, Zhao L, Nurtay A, Abeler-Dörner L, Parker M, Bonsall D, Fraser C (2020) Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science* 368(6491):eabb6936. <https://doi.org/10.1126/science.abb6936>
20. Giannotti F, Nanni M, Pinelli F, Pedreschi D (2007) Trajectory pattern mining. In: Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose, California, USA, August 12–15, 2007, pp 330–339
21. Harvey KL, Zwaan C (1993) Properties and emergence patterns of bipolar active regions. *Solar Phys* 148(1):85–118
22. Howard RF, Sivaraman K, Gupta S (2000) Measurement of kodaikanal white-light images–v. tilt-angle and size variations of sunspot groups. *Solar Phys* 196(2):333–348
23. Huang Y, Shekhar S, Xiong H (2004) Discovering colocation patterns from spatial data sets: a general approach. *IEEE Trans Knowl Data Eng* 16(12):1472–1485
24. Huang Y, Zhang L, Zhang P (2008) A framework for mining sequential patterns from spatio-temporal event data sets. *IEEE Trans Knowl Data Eng* 20(4):433–448
25. Jing J, Yurchyshyn VB, Yang G, Xu Y, Wang H (2004) On the relation between filament eruptions, flares, and coronal mass ejections. *Astrophys J* 614(2):1054–1062. <https://doi.org/10.1086/423781>
26. Kempton D, Angryk R (2015) Tracking solar events through iterative refinement. *Astron Comput* 13:124–135
27. Krucker S, Benz AO, Bastian TS, Acton LW (1997) X-Ray network flares of the quiet sun. *Astrophys J* 488:499–505
28. Langhoff SR, Straume T (2012) Highlights of the space weather risks and society workshop. *Space Weather* 10(6). <https://doi.org/10.1029/2012SW000792>
29. Li Y, Bailey J, Kulik L, Pei J (2013) Mining probabilistic frequent spatio-temporal sequential patterns with gap constraints from uncertain databases. In: 2013 IEEE 13th international conference on data mining, Dallas, TX, USA, December 7–10, 2013, pp 448–457. <https://doi.org/10.1109/ICDM.2013.150>
30. Lloyd's Atmospheric and Environmental Research, Inc. (2013) Solar storm risk to the North American electric grid. <https://www.lloyds.com/news-and-risk-insight/risk-reports/library/natural-environment/solar-storm>
31. LMSAL (2016) Heliophysics Event Knowledgebase. <http://www.lmsal.com/hek/>
32. Louis RE, Kliem B, Ravindra B, Chintzoglou G (2015) Triggering an eruptive flare by emerging flux in a solar active-region complex. *Solar Phys* 290(12):3641–3662. <https://doi.org/10.1007/s11207-015-0726-8>
33. Martens P, Attrill G, Davey A, Engell A, Farid S et al (2012) Computer vision for the solar dynamics observatory (SDO). In: Chamberlin P, Pesnell WD, Thompson B (eds) The solar dynamics observatory. Springer US, pp 79–113. https://doi.org/10.1007/978-1-4614-3673-7_6
34. Mohan P, Shekhar S, Shine JA, Rogers JP (2010) Cascading spatio-temporal pattern discovery: a summary of results. In: Proceedings of the 2010 SIAM international conference on data mining. SIAM, pp 327–338

35. Mohan P, Shekhar S, Shine JA, Rogers JP (2012) Cascading spatio-temporal pattern discovery. *IEEE Trans Knowl Data Eng* 24(11):1977–1992
36. Moore RL, Sterling AC, Hudson HS, Lemen JR (2001) Onset of the magnetic explosion in solar flares and coronal mass ejections. *Astrophys J* 552(2):833–848. <https://doi.org/10.1086/320559>
37. Pei J, Han J, Mortazavi-Asl B, Zhu H (2000) Mining access patterns efficiently from web logs. In: Knowledge discovery and data mining, current issues and new applications, 4th Pacific-Asia conference, PADKK 2000, Kyoto, Japan, April 18–20, 2000, Proceedings, pp 396–407
38. Régnier S, Canfield RC (2006) Evolution of magnetic fields and energetics of flares in active region 8210. *Astron Astrophys* 451(1):319–330
39. Riley P (2012) On the probability of occurrence of extreme space weather events. *Space Weather* 10(2). <https://doi.org/10.1029/2011SW000734>
40. Salas HA, Bringay S, Flouvat F, Selmaoui-Folcher N, Teisseire M (2012) The pattern next door: towards spatio-sequential pattern discovery. In: Advances in knowledge discovery and data mining—16th Pacific-Asia conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29–June 1, 2012, Proceedings, Part II, pp 157–168
41. Schmieder B, Archontis V, Pariat E (2014) Magnetic flux emergence along the solar cycle. *Space Sci Rev* 186(1–4):227–250
42. Shekhar S, Jiang Z, Ali RY, Eftelioglu E, Tang X, Gunturi V, Zhou X (2015) Spatiotemporal data mining: a computational perspective. *ISPRS Int J Geo-Inf* 4(4):2306–2338
43. Solanki SK (2003) Sunspots: an overview. *Astron Astrophys Rev* 11(2–3):153–286
44. Srikant R, Agrawal R (1996) Mining sequential patterns: Generalizations and performance improvements. In: Advances in database technology—EDBT’96, 5th International conference on extending database technology, Avignon, France, March 25–29, 1996, Proceedings, pp 3–17
45. Tian L, Alexander D (2006) Role of sunspot and sunspot-group rotation in driving sigmoidal active region eruptions. *Solar Phys* 233(1):29–43
46. van Driel-Gesztelyi L, Green LM (2015) Evolution of active regions. *Living Rev Solar Phys* 12(1):1
47. Verhein F (2009) Mining complex spatio-temporal sequence patterns. In: Proceedings of the SIAM international conference on data mining, SDM 2009, April 30–May 2, 2009, Sparks, Nevada, USA, pp 605–616
48. Vorpahl JA (1973) Flares associated with EFR’s (emerging flux regions). *Solar Phys* 28(1):115–122
49. Xue C, Liu J, Li X, Dong Q (2016) Normalized-mutual-information-based mining method for cascading patterns. *ISPRS Int J Geo-Inf* 5(10):174
50. Zaki MJ (2001) SPADE: an efficient algorithm for mining frequent sequences. *Mach Learn* 42(1/2):31–60
51. Zhang C, Han J, Shou L, Lu J, Porta TFL (2014) Splitter: mining fine-grained sequential patterns in semantic trajectories. *PVLDB* 7(9):769–780
52. Zhao Y, Zhang H, Cao L, Bohlscheid H, Ou Y, Zhang C (2009) Data mining applications in social security. In: Data mining for business applications. Springer, pp 81–96
53. Zhao L, Chen F, Lu C, Ramakrishnan N (2015) Spatiotemporal event forecasting in social media. In: Proceedings of the 2015 SIAM international conference on data mining, Vancouver, BC, Canada, April 30–May 2, 2015, pp 963–971

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Berkay Aydin, Ph.D is a research assistant professor in the Department of Computer Science at Georgia State University (GSU), as part of the Next Generation—Astroinformatics Research Cluster program. He was a postdoctoral research associate at GSU prior to his current position, where his research was sponsored by NSF and NASA grants. Dr. Aydin’s research is interdisciplinary and focused on management, retrieval, and analysis of Solar Astronomy Big Data. Currently, he works on creating algorithms and data models for multivariate time series prediction and spatiotemporal frequent pattern discovery, which is helpful for understanding the implicit temporal and spatial relationships appearing among objects with spatial and temporal extents as well as forecasting extreme spaceweather events.

Soukaina Filali Boubrahimi is a Ph.D. student at the Computer Science Department, Georgia State University. Her research interests are in the problem of ensemble learning applied to time series data, a common machine learning method used to maximize the learning accuracy and popular in many domains. She is also involved in projects in the data-mining lab consisting of: (1) interpolation of spatio-temporal objects representing solar event trajectories, (2) clustering and visualization of Decision Trees of Coronal Mass Ejection data, and (3) mining discriminative patterns from fMRI-based networks.

Ahmet Kucuk graduated with a M.Sc. degree from Computer Science Department at Georgia State University. His research interests lie in applications of pattern mining from solar event datasets and deep learning from solar image datasets.

Bitá Nezamdoust is a M.Sc. degree at Department of Mathematics and Statistics at Georgia State University. Her research interests lie in development of statistical models for machine learning and data mining applications.

Rafal A. Angryk, Ph.D. received his M.S. and Ph.D. degrees in Computer Science in 2004 from Tulane University. Dr. Angryk also holds two other master's degrees, which he obtained from universities in Poland. In 1999, Dr. Angryk received a M.A. in Business Management from the University of Szczecin and three years later an M.Eng. degree in Computer Science from the Technical University of Szczecin. Dr. Angryk is currently employed as a Professor in the Computer Science Department at Georgia State University (GSU) and came to GSU as part of the university's Second Century Initiative (2CI) to build nationally recognized scholarly strength and critical mass around common research themes. Before joining GSU in August 2013, he spent almost a decade as faculty at Montana State University. Dr. Angryk is the founding director of the MSU/GSU Data Mining Laboratory (dmlab.cs.gsu.edu), and holds the title of Affiliate Professor of Physics & Astronomy at GSU due to the interdisciplinary research he is conducting on massive repositories of solar data. Prior to arriving at MSU, he was involved in the National

Geospatial-Intelligence Agency (NGA) project on Intelligent Database Agents for Geospatial Knowledge Integration and Management, administered by the Center for Computational Sciences in the Stennis Space Center (NASA). Dr. Angryk's research and teaching interests lie in the areas of Very Large Databases (Spatial and Spatio-temporal Databases, and kNN Indexing of High Dimensional Data), Data Mining (Frequent Patterns Discovery, Clustering and Classification of real-life large-scale data), and Information Retrieval (Text and Image data). He has published over 100 journal articles, book chapters, and peer-reviewed conference papers in these areas. His research has been sponsored by several federal agencies: NASA (major contributor), NSF, NGA, as well as industry partners: Intergraph Corporation and RightNow Technologies (now Oracle), with the successful grant history exceeding \$10M.