Finding Sequential Patterns from Massive Number of Spatio-Temporal Events

Yan Huang, Liqin Zhang
University of North Texas
[*huangyan, lzhang*]@cs.unt.edu

Pusheng Zhang
University of Minnesota
[*pusheng*]@cs.umn.edu

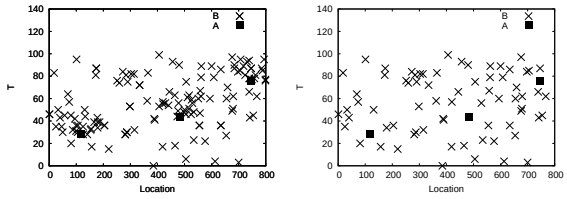
Abstract

Given a large spatio-temporal database of events, where each event consists of the following fields: *event-ID, time, location, event-type*, mining spatio-temporal sequential patterns is to identify *significant* event type sequences. Such spatio-temporal sequential patterns are crucial to investigate spatial and temporal evolutions of phenomena in many application domains. In this paper, we propose a sequence index as the significance measure for spatio-temporal sequential patterns, which is meaningful due to its interpretability using spatial statistics. We propose two algorithms, namely STS-Miner and Slicing-STS-Miner, to tackle the algorithmic design challenges under the spatial sequence index which does not preserve the downward closure property. We evaluate the algorithms by experimentally conducting performance evaluations using both synthetic and real world datasets.

1 Introduction

Many correlated spatio-temporal phenomena tend to exhibit spatial and temporal locality. It is critical to characterize such correlations involving space and time together and identify them from large spatio-temporal datasets efficiently in various application domains. These domains include geographical information science, Earth science, epidemiology, ecology, and climatology [10, 5, 12]. For example, West Nile transmits from one species to another. Birds with the West Nile virus transmit the disease to nearby mosquitoes when they feed on infected birds, then the virus may be injected by mosquitoes into human beings in nearby regions, where it can multiply and possibly cause illness [6]. When many occurrences of such disease transmission from bird to mosquitoes then to human beings are observed, one many conclude the West Nile transmission path as "Bird → Mosquito in nearby region in a day → Human being in nearby region in two days". Here we abstract an event type sequence from their events, and the bird in the pattern represents a phenomenon instead of a particular bird.

We now formally define the problem of mining sequential patterns from spatio-temporal data. Given a database $\mathcal{D}$ of spatio-temporal events, let $\mathcal{F} = \{f_1, f_2, \ldots, f_K\}$ be a set of event types. Each event has an event type and consists of the following fields: *event-id, time, location,* and *event-type*, where *event-type* $\in \mathcal{F}$. We denote the set of events E that are associated with an event type $f \in \mathcal{F}$ as $f.E$. The problem of finding spatio-temporal sequential patterns is to find all the *significant* event type sequences in the form of $f_1 \rightarrow f_2 \ldots \rightarrow f_k$.



(a) (b)

Figure 1: (a) B follows A (b) B is independent of A

1.1 Our Contributions The first challenge in mining spatio-temporal patterns is to identify a statistical *significance* measure to distinguish meaningful patterns from spurious ones. In figure 1, a visual inspection may reveal that B events tend to occur around and after A events in (a), and B events are independently distributed of A events in (b). Quantifying such observation means defining meaningful spatio-temporal statistical significance measures for sequential patterns such as $A \rightarrow B$. The second challenge is to design efficient algorithms to identify significant spatio-temporal sequential patterns from a massive number of spatio-temporal events. A spatio-temporal event type may appear multiple times in a sequential pattern, e.g. $A \rightarrow B \rightarrow A$ is a valid sequence. For K event types and N events, the size of a potential sequential patterns is not bounded by K due to the repletion of event types in sequences. The only limitation is that the sequence is no longer than N since we only have N events. A naive approach to scan the database once for each pattern to check its significance is computationally prohibitive. Downward closure properties are frequently used in data mining algorithms to reduce the searching space, e.g. the Apriori algorithm [1]. However, many meaningful significance measures do not guarantee a downward closure prop-

erty in nature. In this paper, we focus on addressing these two challenges. We make the following contributions: First, we propose a sequence index as the significance measure for spatio-temporal sequences and establish the statistical interpretation using spatial statistics. Second, we propose two algorithms for sequential pattern mining from spatio-temporal data. The proposed algorithms tackle the algorithmic challenges due to the use of the sequence index, which does not guarantee the downward closure property. Third, we experimentally evaluate the performance of two proposed algorithms together using synthetic and real world datasets.

1.2 Related Work Mining sequential patterns from market basket data has attracted much attention since it was introduced by Agrawal *et al.* [2]. It addressed the following problem. Given customers' purchasing records, the objective is to find common purchased subsequences shared by significant number of customers, e.g., (PC, Internet service) $\rightarrow$ DVD $\rightarrow$ MP3 player. Many efficient algorithms [9, 11] have been proposed to identify sequential patterns from the market basket data. However, the "transactionization" of spatio-temporal space to adapt the existing sequential pattern mining methods on market basket data is a non-trivial task and may be un-natural due to the continuity of spatial space and time [7].

The trajectory of a moving object is typically a collection of spatial signatures at consecutive time stamps. Retrieving similar trajectories might reveal underlying traveling patterns of moving objects in the data. Mamoulis *et. al.* discussed mining, indexing, and querying of historical spatio-temporal data in [8, 3]. In their context, trajectories of objects are given for investigations. This work is related but different from our work. Trajectory is the collection stops of the **same** moving object at different spatial locations. Trajectory analysis can be applied only if the trajectories have been provided a priori. In our context of spatio-temporal sequential pattern mining, the spatio-temporal data is the collection of different events, with each event belonging to one particular event type as shown in Figure 1. So trajectory may not be available for spatio-temporal sequential patterns in this context.

2 Significance Measure

We define significance measure in this section. For simplicity of illustration, we assume that the dimensionality of both space and time is one in this paper. All the definitions and algorithms can be generalized into higher dimensions for space and time as well.

We say an event e' *follows* an event e, denoted by $e \rightarrow e'$, iff e' is in the spatio-temporal neighborhood of e.

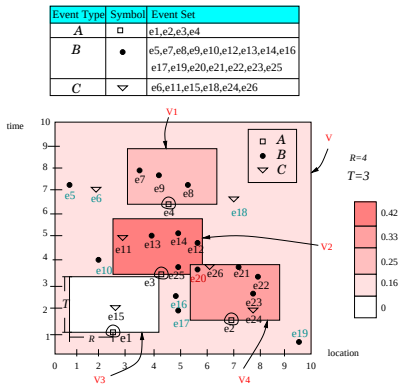


Figure 2: Concept Illustration: Density of B events in different regions

A neighborhood is user given and application specific. An example neighborhood could be defined as a distance less than 1.5 miles for any time interval less than 1 hour. In Figure 2, three spatio-temporal event types are represented using the symbols square (A), circle (B) and triangle (C) respectively. Their events are embedded in a spatio-temporal space. The x-axis denotes the linear space and the y-axis denotes the linear time. The event set for each event type is listed in the table in the same figure. Among many others, the event e_{11} follows the event e_3, i.e., $e_3 \rightarrow e_{11}$ for a given distance $\mathcal{R}$ and time interval $\mathcal{T}$ as show in the figure.

One way to define the significance measure of a pattern $f_1 \rightarrow f_2$ is to calculate the number or density of type f_2 events that follow type f_1 events. However, this approach ignores the distribution of f_2 events in the overall space. For example, in Figure 1 (b), when event type B has many events, this naive significance measure will wrongly conclude that since there are many B events following A events (or the density of B events following A events are high) then $A \rightarrow B$. Instead of simple counting, we propose a sequence index as a significance measure for spatio-temporal sequential patterns, which may be more meaningful due to its interpretability using spatial statistics [4].

2.1 Density Ratio for Two Event Sets In order to define density ratio, we introduce *density* first. For a given spatio-temporal region V, the density $density(E, V)$ of an event set E in V is the average number of events in event set E in each unit of V, i.e. $density(E, V) = \frac{|\{e|e \in E \wedge e \in V\}|}{|V|}$, where $|V|$ represents the volume of the region V. In Figure 2, the density for event type B in the whole spatio-temporal embedding space V is $\frac{16}{|V|} = 0.16$ since $|V| = 10 \times 10 = 100$, which means in each unit of the spatio-temporal embedding space, we expect to find 0.16 type B events. The density for event type B in region V_1 is $\frac{3}{|V_1|} = 0.25$, where

$|V_1|$ is the volume of neighborhood of e_4, given $\mathcal{R} = 4$ and $\mathcal{T} = 3$, $|V_1| = 4 \times 3 = 12$. To visually represent the density concept, the densities for event type B in the whole space V and in each A event's neighborhoods $V_1, \ldots, V_4$ are computed and represented with different shades of gray in Figure 2.

Given two event sets E and E', if the density of E' around events of E is higher than the density of E' in the overall embedding space, then it is likely that E' events tend to follow E events. In Figure 2, the density $density(B, V)$ of type B events in the overall embedding space V is 0.16 as calculated before. The average density of B events around A events is $\frac{0+\frac{4}{12}+\frac{5}{12}+\frac{3}{12}}{4} = 0.25$. Compared with the overall density of B in the whole embedding space, the density ratio of these two densities is $\frac{0.25}{0.16}$ which is greater than 1. This means $A \rightarrow B$. Let us define the density ratio concept formally as follows:

DEFINITION 2.1. (DENSITY RATIO) *For two event sets E and E' and a given neighborhood function $N_{\mathcal{R}}^{\mathcal{T}}(e)$, the density ratio of $E \rightarrow E'$ is defined as: $densityRatio(E \rightarrow E') = \frac{average_{e \in E}(density(E', N_{\mathcal{R}}^{\mathcal{T}}(e)))}{density(E', V)}$ where V is the spatio-temporal embedding space.*

2.2 Sequence Index A spatio-temporal sequence of k event types is called a (k)-sequence. We refer to the $(k-1)$-subsequence consisting of the first $(k-1)$ event types of a (k)-sequence S as $S[1 : k-1]$ and the i^{th} event type in a sequence S as $S[i]$. Generalizing density ratios to longer-sequences $(k > 2)$ is non-trivial. In order to keep a sequence S to be significant, there are two properties that significance measures need to observe:

1. The events of any event type $S[i]$ in S need to "$follow$" all of the preceding event types of sequence $S[1 : i-1]$
2. The density ratio between any two consecutive event types needs to be significant.

To achieve the first property, the tail event set of a (k)-sequence needs to be defined to capture the $follow$ relationship between an event type and a sequence. The tail event set is a recursively defined concept. For 1-sequence, the tail event set is simply the event set of the event type in the singleton sequence. For a (k)-sequence, it is defined based on the tail event set of its prefix $(k-1)$-sequence. The tail event set of a (k)-sequence include those events that follow the tail event set of its prefix $(k-1)$-sequence. It is a subset of events of event type f_k.

We formally define the tail event set of a sequence as follows:

DEFINITION 2.2. (TAIL EVENT SET OF A SEQUENCE) *The tail event set of a (k)-sequence S is defined recursively:*

1. *When $k = 1$, $tail\_event\_set(S)$ of a (1)-sequence S is defined as: $tail\_event\_set(S) = S[1].E$;*
2. *When $k \geq 2$, $tail\_event\_set(S) = \{e | e \in S[k].E \wedge \exists e' \in tail\_event\_set(S[1 : k-1])(e' \rightarrow e)\}$*

To achieve the second property, we define a sequence index to be the minimal of the density ratio between the tail event set of an $(i-1)$-subsequence $S[1 : i]$ and the event set of the event type $S[i]$, where $i \in [1, k-1]$. Instead of using all of the events in $S[i-1].E$, we use the tail event set of $S[1 : i-1]$ to compute the density ratio with $S[i].E$ because we want to measure the impact of the $(k-1)$-sequence on the event type $S[i]$, instead of the impact of event type $S[i-1]$. Now we use density ratio and tail event set to define sequence index:

DEFINITION 2.3. (SEQUENCE INDEX) *The sequence index of a (k)-sequence S is defined as:*

1. *When $k = 2$, $seqIndex(S) = densityRatio(S[1].E \rightarrow S[2].E)$;*
2. *When $k \geq 3$ $SeqIndex(S) = \min(seqIndex(S[1 : k-1]), densityRatio(tail\_event\_set(S[1 : k-1]), S[k].E))$*

We say a (k)-sequence S a *significant* spatio-temporal sequence if $seqIndex(S) \geq \theta$, where θ is a user given minimum sequence index. *The problem of mining spatio-temporal sequential patterns is to find all significant spatio-temporal sequences.*

LEMMA 2.1. *Sequence index is not anti-monotone and does not guarantee the downward closure property for the sequential patterns.* [1]

3 Mining Spatio-temporal Sequential Patterns

In this section, we propose two algorithms for sequential pattern mining from spatio-temporal data. The proposed algorithms tackle the algorithmic challenges using the sequence index, which does not guarantee the downward closure property. The first algorithm STS-Miner is an iterative pattern growth algorithm. When the number of events are too large to be processed in memory, we propose the second algorithm: Slicing-STS-Miner.

[1] The proof is omitted due to space limitation.

3.1 STS-Miner The algorithm STS-Miner starts off with all singleton sequences and generates candidate $(k+1)$-sequences by attaching one more event type to significant (k)-sequences.

A pattern forest consisting of trees starting with each event type is maintained. Each node of a tree represents a sequential pattern starting from the root and has the information about the last event type of the sequential pattern it represents and its tail event set. To create the pattern forest, we create a tree starting with each event type and expand it using a depth first procedure. For a given node, the procedure tries every event type in the database to expand the current node. For each trial, we compute the sequence Indies and the tail event set. If the computed index is above the threshold, a new node with the current event type is added and will be recursively expanded further.

3.2 Slicing-STS-Miner The Slicing-STS-Miner algorithm has three phases: hashing phase, mining and merging phase, and pruning phase. In the hashing phase, we divide the time dimension into overlapping slices. Every two consecutive slices overlap by $\mathcal{T}$, where $\mathcal{T}$ is the given time interval for *follow* relationship. We *hash* each event into slices by its time stamp. One event could be hashed into two slices if its time stamp is in the overlapping area of two slices. This overlapping area is used to "stitching together" event sequences that cross boundaries. In the mining and merging phase, we process the slices in a time increasing order while updating a single forest of pattern trees. When we process a slice, we use the depth first expanding procedure as described before to generate all the sequential patterns. The θ is set to 0, since the computed sequence index is only for slices processed so far instead of for whole dataset. We have to wait until we process all the slices and then do the pruning using the minimum sequence index threshold.

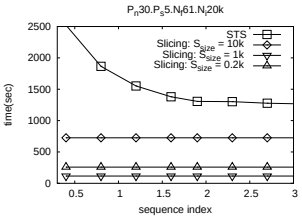
4 Experimental Design and Results

We conducted an extensive performance study on both synthetic and real datasets. All experiments are performed on a Pentium IV 3.2 GHZ PC machine with $1G$ megabytes main memory, running Debian GNU/Linux. All algorithms are implemented using Java.

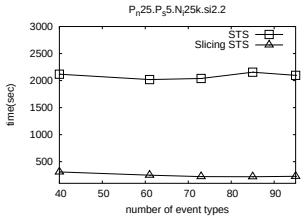
4.1 Experiments on Synthetic Datasets Synthetic data are generated using a data generator similar to the one by Agarwal et al. [1], along with an extension to produce spatial and temporal datasets. The important parameters are explained in Table 1. In our experiments, the space and time framework is set to $100 \times 1000 \times 100$. Following are some additional default values when not specified: $\mathcal{R} = 5, \mathcal{T} = 5$, and $N_f = 100$.

Table 1: Parameters of Data Generator

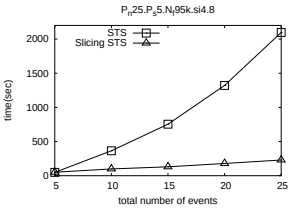
Parameter	Definition
p_n	number of max sequential patterns
p_s	mean size of max sequential patterns
N_f	number of event types participating in a pattern
N_n	number of noise events
N_i	total number of events
$\mathcal{R}$	maximal distance two event related
$\mathcal{T}$	maximal time interval two events related
s_{size}	the size of each slice
θ	Minimum index threshold



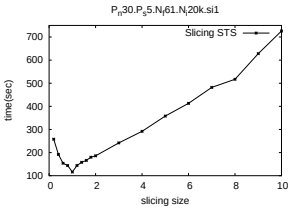
(a) Performance w.r.t Sequence Index



(b) Performance w.r.t. Number of Event Types Participating in a Pattern



(c) Performance w.r.t. Number of Events



(d) Performance w.r.t Slicing Size

Figure (a) shows the performance on a dataset with $20k$ events and 61 event types with respect sequence index thresholds. The Slicing-STS-Miner outperforms the STS-Miner by a factor of 10 - 22 when the slice size is chosen properly. With the sequence index decreasing, the performance of STS-Miner decreases exponentially, while the speed of Slicing-STS-Miner remains the same.

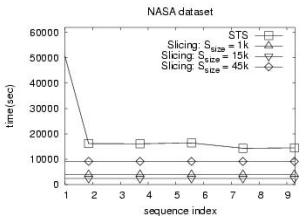
Figure (b) shows the performance of the algorithms with $25k$ events w.r.t the number of event types participating in a pattern increasing on a dataset of $25K$ events. Both algorithms are stable with the increasing number of event type due to the same number of events. This implies both algorithms are not sensitive to the number of event types.

Figure (c) shows the performance with 95 event

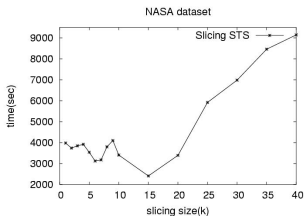
types w.r.t. the total number of events. With the number of events increasing, the running time is also increasing for both algorithms. However, the STS-Miner slows down exponentially, while the Slicing-STS-Miner slows down linearly. For this data set, the Slicing-STS-Minner algorithm outperforms STS-Miner with an improvement factor up-to 9.

Figure (d) shows the performance of Slicing-STS-Miner with a increasing slice size. Note that Slicing-STS-Miner is not sensitive to the sequence index threshold θ. As one may notice, the performance increases first and then decreases. The reason may be that small slicing introduces lots of overheads in "stitching together" sequences across slicing boundaries and too large slicing may have a memory problem when processing each slice.

4.2 Experiments on Real Datasets Earth Science data consists of a sequence of global snapshots of the Earth taken at various points and time [12]. In our experiments, we choose to use the data set available from University of Minnesota [2]. A subset of the real data was used. It includes the highSolar, highPET, higTempave and lowPrec, and their events. We choose the first 25 months of the data and fix the spatial neighborhood $\mathcal{R} = 5$ half degrees and the time interval $\mathcal{T} = 5$ months.



(e) Performance w.r.t. Sequence Index Thresholds



(f) Performance w.r.t. Slicing Size

Figure (e) illustrates the performance of the mining algorithms with respect to density index thresholds. With properly selected slicing size, the Slicing STS-Miner outperforms STS-Miner with a factor between 7 and 24, which is very similar to Figure (a).

Figure (f) shows the performance of the Slicing-STS-Miner with respect to slicing size. The performance has slight fluctuation in the early part of curve. The overall performance of the algorithm reaches its best when each slice contains approximately 15K events and then the performance decreases.

[2] Discovery of Changes from the Global Carbon Cycle and Climate System Using Data Mining, http://www.ahpcrc.umn.edu/nasa-umn

5 Conclusion and Future Work

In the paper, we formally defined the problem of mining spatio-temporal sequential patterns. We proposed a sequence index as the significance measure for spatio-temporal sequences and established the statistical interpretation using spatial statistics. We proposed two algorithms for sequential pattern mining on spatio-temporal data. The proposed algorithms tackled the algorithmic challenges using the sequence index, which did not guarantees the downward closure property. We conducted performance evaluations for the proposed algorithms using both real and synthetic datasets. Further research may include studying the effects of using variable spatio-temporal neighborhood instead of constant ones as we assumed in this paper.

Acknowledgement: We are very grateful to Dr. Vipin Kumar and Mr. Michael Steinbach at the University of Minnesota for providing the climate data for our experimental evaluations.

References

[1] R. Agarwal and R. Srikant. Fast Algorithms for Mining Association Rules. In *Proc. of VLDB*, 1994.

[2] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Proc. of SIGKDD*, 1995.

[3] H. Cao, D. W. Cheung, and N. Mamoulis. Discovering partial periodic patterns in discrete data sequences. In *Proc. of PAKDD*, 2004.

[4] N.A.C. Cressie. *Statistics for Spatial Data*. Wiley and Sons, ISBN:0471843369, 1991.

[5] Manolis Koubarakis et. al. *Spatio-Temporal Databases: The CHOROCHRONOS Approach*. Springer, 2003.

[6] Centers for Disease Control and Prevention (CDC). http://www.cdc.gov/ncidod/dvbid/westnile.

[7] Y. Huang, S. Shekhar, and H. Xiong. Discovering Co-location Patterns from Spatial Datasets:A General Approach. *IEEE TKDE*, 16(12), 2004.

[8] N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. L. Cheung. Mining, indexing, and querying historical spatiotemporal data. In *Proc. of SIGKDD*, 2004.

[9] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto J. Wang, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *Proc. of SIGKDD*, 2004.

[10] J.F. Roddick and M. Spiliopoulou. A Bibliography of Temporal, Spatial and Spatio-temporal Data Mining Research. *ACM SIGKDD Exploration*, 1999.

[11] M. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

[12] P. Zhang, M. Steinbach, V. Kumar, S. Shekhar, P. Tan, S. Klooster, and C. Potter. Discovery of Patterns of Earth Science Data Using Data Mining. In *Next Generation of Data Mining Applications*. 2004.