1 Problem statement

In this paper, authors (Pradeep et al.) have addressed the problem of finding partially ordered sub-sets of the events from the given collection of boolean spatio-temporal (ST) event-types in such a way that the sub-set contains events which are located together (spatial component) and occur serially (temporal component). These partially ordered sub-sets are called Cascading Spatio-Temporal Pattern (CSTP). The motivation behind Discovering CSTPs from ST datasets is its application in domains such as public safety, climate modelling and natural disaster planning etc. For example in the domain of public safety, if we create CSTPs of crime datasets for the analysis and can reveal some pattern like drunk driving occur near and after bar closing, and increases on weekends or near large gatherings like game nights, law enforcement agencies can schedule patrolling on such places/events and increase public safety.

As the authors have presented, "the CSTP mining problem can be defined formally as follows:

**Given:** (a) a ST dataset consisting of a set of Boolean event-types over a common ST framework; (b) a directed neighbour relation $R$; and (c) a threshold for Cascade Participation Index (CPI).

**Find:** CSTPs with CPI $\geq$ the user specified threshold."

(The directed neighbour relation $R$ and CPI is explained in section 3.)

In this paper, the interpretation of term ST autocorrelation is different from the statistical correlation terms such as cross correlation. Here positive ST autocorrelation is used to refer clustering of event-type instances within a ST framework.

The key challenges in discovering CSTP from ST datasets, listed by author, are:

1. With the increase in number of event-types, the cardinality of candidate space increases exponentially. This in turn increases the computational complexity of the problem.
2. enumeration of ST neighbourhood for interest measure computation is a complex problem.
3. computational scalability and statistical interpretation have conflicting demands i.e. polynomial time computable measures gives computation scalability but does not guarantee to give measures that can detect ST correlation.

2 Major contributions

Authors have extended their preliminary work on CSTP [1] as part of this paper. Recent works of the authors [1, 2, 3] are also included as part of this paper. The contribution listed in this paper as part of previous works are as follows:

1. A novel interest measure for CSTPs to qualify for statistically meaningful CSTP.
2. A CSTP miner (CSTPM) based on simple loop to evaluate interest measure. This is called SIAM NL-CSTPM.
3. Two filtering strategies called *Upper Bound (UB) filter* and *Multi-resolution spatio-Temporal (MST) filter* to eliminate non-prevalent patterns during discovering CSTPs from ST datasets.

4. Analysis of real crime datasets of Lincoln, Nebraska to show interesting CSTPs.

The major contributions of these paper, which are new in this paper and extends the previous works, are as follows:

1. A bottleneck analysis of computation time for the CSTPM. The results show that major chunk of computation time is taken by interest measure evaluation compared to candidate generation.

2. A new algorithm to compute interest measures (called TKDE STP-CSTPM). This algorithm is based on ST partitioning. Correctness and completeness of this new algorithm for CSTPM is proved.

3. An algebraic cost model for the filtering strategies proposed in previous works.

4. Experimental comparison between SIAM NL-CSTPM and TKDE STP-CSTPM on synthetic and real datasets to show that later one outperforms the former one.

5. Generalization of the experimental results for evaluating the proposed filtering strategies on synthetic datasets. Also a new synthetic data parameter (called clumpiness degree) is included to control the extent of positive ST autocorrelation. Evolution of filters on large datasets is also shown.

6. Case study of the previous work has been extended to find CSTPs from real datasets to illustrate the new algorithm. A revised view of CSTPM process and CSTP candidate space is also presented for the new algorithm.

### 3 Key concepts and examples

Before going into the key concepts of the proposed algorithm (TKDE STP-CSTP) author has explained some basic concepts related to modelling ST data and interestingness of CSTPs, which are necessary for understanding the solution. These concepts are as follows (as author has presented):

"**Direct neighbour relation (R):** A direct neighbour relationship over a set $MI$ of even instances may be formally represented as a directed acyclic graph (DAG), where $EI$ is a set of directed edges representing ordered pairs in $MI \times MI$ (please refer fig 3. in the paper for an example of directed neighbour relation). The edges in the graph are computed using CSTPM."

"**Measure of interestingness of CSTPs:** In the context of ST data mining, the measure of interestingness is balanced between computational scalability and statistical interpretation. A key application domain of CSTP requires an ability to predict the instance of CSTP on the occurrence of a participating event type. These requirements create a need of interest measure based on conditional probability. The author has presented following for the measure of interestingness of CSTPs:

1. **Cascade Participation Ratio**, $CPR(CSTP, M)$: It is the estimate of conditional probability of an instance of CSTP given an instance of event type $M$, i.e. $Pr(CSTP|M)$. Formally $CPR(CSTP, M)$ can be written as:

$$CPR(CSTP, M) = \frac{\#\text{instances}(M) \text{ participating in CSTP}}{\#\text{instances}(M) \text{ in dataset}}$$
2. **Cascade Participation Index, CPI(CSTP):** It is a measure of the likelihood of an instance of CSTP in the ST neighbourhood of an instance of participating event-type. So by definition, CPI(CSTP) is defined as the minimum of CPR(CSTP,M) over all event-types in a CSTP.

\[
CPI(CSTP) = \min\{CPR(CSTP,M)\}
\]

So in another way, CPI(CSTP) can also be viewed as a lower bound on the conditional probability \(Pr(CSTP|M)\) for any event type M.

To reduce the computation complexity due to exponential candidate space, author has explained two filtering strategy as shown below:

1. **Upper bound (UB) filter:** The UB filter is based on the existence of an upper bound for the CPI, which is the ratio of the minimum and maximum value of CPR of participating event-types, i.e. for an event \(M\) participating in a CSTP

\[
upper(CPI) = \frac{\min\{CPR(CSTP,M)\}}{\max\{CPR(CSTP,M)\}}
\]

These upper bound values reflect the maximum possible value of the interest measure and can be used to filter out the candidates with low interest measure values. These uninteresting candidates can potentially slow down the computation. This UB filter is used before the candidate generation and this way reduces the candidate set for which interest measure need to be computed.

2. **Multi-resolution Spatio-Temporal (MST) filter:** Event instances of the same event type can clutter together due to the presence on positive ST autocorrelation in ST frameworks. The MST filter exploits this property of geographic clustering of event instances to compute a coarse CPI by creating coarse direct neighbour relation \(R^C\) from the cluttered ST dataset. Authors have given the proof that this coarse CPI will never underestimate the CPI (please refer the original paper). This MST filter is used before the actual CPI computation as actual CPI computation is computationally expensive.

Using the above described concepts and filtering technique, author has presented an algorithm for CSTPM. The important steps of which is as follows:

For each size \(k\) of pattern, 1) apply UB filter; 2) generate candidate of size \(k\) using CSTPs of size \((k-1)\); 3) perform MST filtering to prune out non-prevalent patterns; 4) Generate pattern instances and compute actual CPI; 5) prune CSTPs based on their prevalence; 6) Generate prevalent CSTPs. (please refer original paper for details)

### 4 Validation methodology

Author has given mathematical proof for the completeness and correctness of the CSTPs generated by new algorithm. Also various theorems and lemma are presented to provide mathematically proof for the filtering strategy. These mathematical proofs and theorems are very useful to understand and gives confidence on the correctness and completeness of the algorithm without looking at the experimental results.
In addition to the mathematical proofs, authors have given experimental results to 1) compare the performance of the two different CSTPM algorithms called SIAM and TKDE without applying filtering and 2) to determine the impact on computation performance of TKDE if filtering is applied. The authors have performed multiple experiment on both synthetic and real world datasets to show the effect on computational complexity with the variation of dataset size, number of event types, use of filtering etc. Also authors have used the clumpiness parameter to control the clustering to show the impact of clustering on ST join operation. In addition the these experiments, the authors have presented a case study of the crime data of Lincoln, Nebraska. The statistical interpretation of the CSTPs created by crime data of Lincoln, Nebraska has been presented as part of it.

5 Assumptions and limitations

The authors have made an assumption about the type of Spatio-temporal event. In this paper, authors have considered that Spatio-temporal events are of Boolean type. We think that this assumption is valid and in accordance with the problem described in this paper. The CSTPs shows partial order of events which occur in close vicinity and serially and also it is primarily concerned with the occurrence and absence of a event type at a particular location and time. So the assumption of boolean event type is sufficient for discovering CSTPs and provide statistical interpretation based on it.

In addition to above assumption, the authors have limited the scope of this paper as follows:

1. The focus of this paper is on the computational aspects of CSTP discover. The problems of ST non-stationarity, choices of directed neighbour relationships and interest measure or the motion of ST instances are not in the scope this paper.

2. Grid cell size, which are crucial to MST filter performance, is also outside the scope of this paper. The grid cell sizes are dependent on application domain. So cannot be generalized and hence beyond the scope of this paper.

6 Critique

The paper in well written but have one minor printing issues. While we were reading the paper we noticed that the example shown in fig 6 is not in accordance with the original example fig 3 (please see the original paper). Also the values of the table in fig 6 have some printing errors. We have communicated this with the author of this paper and he has acknowledged the mistake.

Apart from the above printing mistake, this paper have few limitations which can be improved. These limitation are listed below:

1. The authors have considered Euclidean distance, which is not a good measure for the distance for real world problems. It would be good to improve the approach to consider actual network distance instead of euclidean distances and see the performance on a real world datasets.

2. The authors have described the filters (MST and UB filters) and shown it’s impact on the computation complexity of algorithm. But they have not shown the impact of individual filters and significance of one in the absence of other. This experiment would have proved the claims about each filter made by the authors experientially.
3. The authors have used brute force approach for candidate generation, which can be improved further to reduce the computational complexity.

References

