The University of Osaka
Institutional Knowledge Archive

Title	Causal inference on spatio-temporal point process data based on Granger Causality
Author(s)	Pavasant, Nat
Citation	大阪大学, 2024, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/96232
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Causal inference on spatio-temporal point process data based on Granger Causality

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2024

Nat PAVASANT

Publications

Journal

1. Nat Pavasant, Takashi Morita, Masayuki Numao, Ken-ichi Fukui. Granger causality-based cluster sequence mining for spatio-temporal causal relation mining. *International Journal of Data Science and Analytics* (2023). https://doi.org/10.1007/s41060-023-00411-x

2. Nat Pavasant, Takashi Morita, Masayuki Numao, Ken-ichi Fukui. Local density estimation procedure for autoregressive modeling of point process data. *The IEICE Transactions on Information and Systems* (under review)

International Conference

1. Nat Pavasant, Hiroshi Furutani, Masayuki Numao, and Ken-ichi Fukui. “ART-2b: Adapted ART-2a for large scale data clustering on PM2.5 mass spectra”, *Proc. 2017 IEEE International Conference on Big Data (IEEE BigData)*, pp. 4813-4815, 2017. (poster)

2. Nat Pavasant, Masayuki Numao, and Ken-ichi Fukui. “Spatio-Temporal Change Detection Using Granger Sequence Pattern”, *Proc. the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI2020)*, Doctoral Consortium, pp. 5202-5203, 2020.

3. Nat Pavasant, Masayuki Numao, and Ken-ichi Fukui. “Spatio-Temporal Change Detection with Granger Causality Based Cluster Sequence Mining”, *Proc. 19th IEEE International Conference on Machine Learning and Applications (ICMLA2020)*, pp. 551-558, 2020.

Domestic Conference

1. Nat Pavasant, Masayuki Numao, Ken-ichi Fukui. “Spatio-temporal Change Detection Using Pattern Time Signature”, The 118th JSAI SIG-KBS, 2019.

2. Nat Pavasant, Masayuki Numao, Ken-ichi Fukui. “Spatio-Temporal Change Detection Using Granger Causal Relation”, The 34th Annual Conference of the Japanese Society for Artificial Intelligence, 2020. (JSAI Annual Conference Award, International Session Oral Presentation)

3. Nat Pavasant, Takashi Morita, Masayuki Numao, Ken-ichi Fukui. “A kernel-density-based procedure for autoregressive modelling of point process data”, The 130th JSAI SIG-KBS, 2023.

Abstract

Real-world data are usually in the form of spatio-temporal data, such as weather systems, transport demand, and disease outbreaks. However, the complexity of this type of data means that analysis techniques are not as well established. Identifying relationships, specifically causal relationships, within the spatio-temporal data can yield further understanding of natural phenomena, but the area is not well understood. This work proposed a method to extract causal relations of clusters from multi-dimensional event sequence data. The proposed Granger Cluster Sequence Mining (G-CSM) algorithm identifies the pairs of spatial data clusters that have causality over time with each other. It extended the Cluster Sequence Mining algorithm, which utilized a statistical inference technique to identify occurrence relation, with a causality inference based on Granger causality. In addition, the proposed method utilizes a false discovery rate to control the significance of the causality. The method was tested using both synthetic data and semi-real data and can extract embedded causal relations with high F-scores over different sets of data even under high spatial noise. False discovery rate also helps to increase the accuracy even more under such cases and also makes the algorithm less sensitive to the hyper-parameters. Furthermore, a local density estimation procedure was also proposed. This procedure is a pre-processing step to the vector autoregressive modeling of point-process data, a process which was used during Granger causality inference in the proposed algorithm, by applying a density estimation. Results on synthetic data showed that the procedure improved model accuracy, especially under sparse data.

Acknowledgement

First and foremost, I would like to express my sincere gratitude toward Professor Masayuki Numao for his warm welcome and the opportunity to be a part of his laboratory.

Second, I would like to express my gratitude and appreciation toward Associate Professor Ken-ichi Fukui, my adviser, for his tremendous support and advice during my time under his supervision. His patience, encouragement, and advice are what allowed this thesis to be completed. I would also like to express my gratefulness to Assistant Professor Tsukasa Kimura and Assistant Professor Takashi Morita for beneficial opinions and discussions during my time.

Next, I would like to thank all of my friends, seniors, and juniors in the Numao Laboratory and at Osaka University, for providing a wonderful atmosphere to conduct research, for helping me through difficult times, and for any advice you may have given me.

Lastly, I would like to thank my family who supports me, listens to my problems, and always offers encouragement during my studies. Especially my farther, who have shown me the way of academics since I was young, and has sadly passed away before I could show him my doctoral dissertations.

Contents

List of Figures v

List of Tables vii

1 Introduction 1

2 Literature review 8

2.1 Spatio-temporal point process 8

2.1.1 Intensity Function 8

2.1.2 First-order Separability 9

2.1.3 Spatial Cluster with Temporal Relation 10

2.2 Relation mining 10

2.3 Causal Inference 11

2.3.1 Granger Causality 12

2.3.2 Transfer Entropy 14

2.3.3 Constraint-Based Algorithms 14

3 Granger Cluster Sequence Mining (G-CSM) 16

3.1 Overview 16

3.2 Cluster Sequence Mining (CSM) 16

3.3 Granger Causality 21

3.3.1 Significance testing 24

3.4 Granger Cluster Sequence Mining (G-CSM) 25

3.4.1 Time Proximity Evaluation 25

3.5 Experiments 26

3.5.1 Data generation 26

3.5.2 Evaluation measure 28

3.5.3 Performance validation 29

3.5.4 Parameter analysis 35

3.5.5 Other type of patterns 37

3.5.6 Semi-real data 38

3.5.7 Complexity Analysis of the G-CSM algorithm 40

3.5.8 Limitation of the G-CSM algorithm 41

3.6 Summary 42

4 Local Density Estimation for Point Process Vector Autoregressive Model 43

4.1 Overview 43

4.2 VAR modeling of point process 44

4.3 Local density estimation 46

4.4 Experiments 51

4.5 Complexity analysis 52

4.6 Summary 53

5 Conclusion 54

5.1 Summary 54

5.2 Future Works 55

5.2.1 Applying Local Density Estimation to the G-CSM algorithm . . 55

5.2.2 Spatial Clustering Improvement on G-CSM algorithm 55

5.2.3 Extension to Non-Stationery Relations 55

5.2.4 Non-Linear Extension fo Causality Detection 56

5.2.5 Performance Improvement to the G-CSM algorithm 56

5.2.6 Other time interval distribution 56

5.2.7 Application to real-world usage 57

List of Figures

1.1 Point process and spatio-temporal point process data 2

1.2 Different types of causal relations. 3

1.3 Correlation between spatial clusters of earthquake occurrence as extracted by Cluster Sequence Mining algorithm. 5

1.4 Concrete problem definition 6

2.1 Network relationship discovery from fMRI data. 11

2.2 A simplified view of Granger Causality model 13

3.1 Overview of CSM algorithm 18

3.2 AHC dendrogram nodes 18

3.3 Simplified view of the GLM model for point-process Granger causality. . 22

3.4 Example of the input data 28

3.5 Example of relation(s) extracted by each algorithm 32

3.6 Scatter plot of the temporal and spatial score. 34

3.7 Histogram of $\mathcal{L}$, CSM vs G-CSM. Noise = 100, $\lambda = 2$. 36

3.8 Generated data and extracted relations using G-CSM with FDR-SC for other types of pattern 39

3.9 Semi-real experiments data detail 40

4.1 A model using local density estimation 44

4.2 Local Density vs Global Density 46

4.3 Difference between history length and number of parameters. 47

4.4 Histograms of the interval between events at different sparsity. 48

4.5 Mean-squared error (MSE), log-likelihood, and F1-score of each model. . 49

List of Tables

1.1 A brief comparison of different spatio-temporal relation mining methods 4

3.1 AIC value at different history length. Bold is the lowest. 30

3.2 CSM and G-CSM at various noise levels. 31

3.3 CSM and G-CSM at various relation intervals. 33

3.4 Evaluation scores of the relation with the highest evaluation score from G-CSM 35

3.5 CSM and G-CSM at various alpha settings. 37

3.6 CSM and G-CSM result from digits dataset. 40

4.1 Models used in the experiment 48

4.2 Mean-squared error (MSE), log-likelihood, and F1-score of each model. . 50

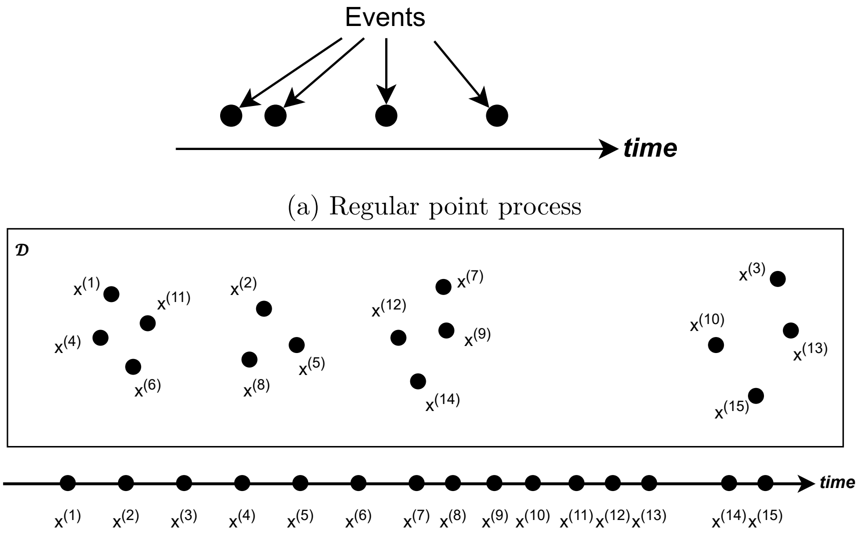
Chapter 1

Introduction

Many of the data being generated today are spatio-temporal in nature. Many real-world organizations deal with a large amount of spatio-temporal data on a day-to-day basis. Such organizations are spread across multiple fields including aerospace, meteorological, transportation, police, and healthcare [1]. The applications also range widely including the ecology and environmental management [2], crime analysis [3], transport route analysis [4], disease management [5], precision agriculture [6], and many more.

Spatio-temporal data have both a spatial part and a temporal part. They represent multidimensional, continuous data at a specific time point. This is in contrast to spatial data, which are just data points in some n-dimensional space. They are also different from temporal data, which are data over a span of time. At present, there are still many challenges in the analysis of spatio-temporal data. Because they have both spatial and temporal parts, spatio-temporal data are inherently more complex than just spatial data or temporal data, with data and relationships that may span across spatial and temporal domains. Existing techniques for spatial data or temporal data do not work well with spatio-temporal data. Moreover, precisely because the spatio-temporal data span across many data domains, domain knowledge is also required [7]. Therefore, new techniques are being researched and developed specifically for spatio-temporal data and each problem domain [1] to extract knowledge from a wealth of spatio-temporal databases.

A *spatio-temporal point process* [8] is a type of point process data (Fig 1.1a). A regular point process is a list of timestamps, or events, on a timeline. That is, the



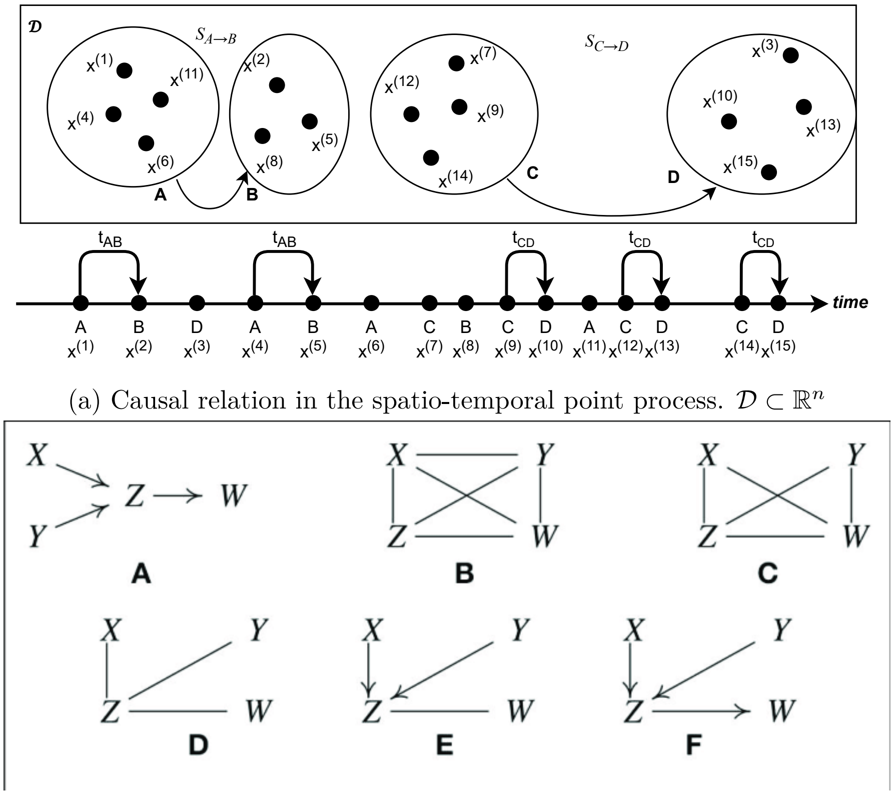
(a) Regular point process

(b) Spatio-temporal point process. $\mathcal{D} \subset \mathbb{R}^n$

Figure 1.1: Point process and spatio-temporal point process data

data itself is the timestamp. This is in contrast to time series where the timestamps have constant intervals with the actual data being the value at each timestamp. When each event has an associated value, that is both the timestamp and the value are the data, this becomes a spatio-temporal point process. This is illustrated in Fig. 1.1b. A spatio-temporal point process can represent a pinpoint spatial location on a sparse, unscheduled timeline. An example of a spatio-temporal point process includes a list of points in an Euclidean space, etc. This type of data has freedom in both temporal and spatial dimensions. Many real-world spatio-temporal data can be represented as a point process, such as earthquake epicenters as a list of latitude and longitude as the spatial part and the occurrence time as the temporal part; or social network posts can be considered as features extracted by natural language processing (NLP) algorithm for the spatial part and the post time as the temporal part.

Existing research using point-process spatio-temporal data includes modeling earthquake [9] or ambulance demand [10]. A direct model of a spatio-temporal point process is very hard to optimize, resulting in various model simplifications [11]. Newer developments included neural-network-based method [12] or using reinforcement learning [13]. These works focused on modeling the actual point process, which is a mathematical model that captures the occurrence of each data point in the spatio-temporal domain. This model is useful for studying the mechanics of each occurrence or for predicting future events, however, none of the existing methods deal with relation extraction. Since a



(a) Causal relation in the spatio-temporal point process. $\mathcal{D} \subset \mathbb{R}^n$

(b) A PC algorithm finding causal relations within random variables. A is the true relations, with the PC algorithm iterating to find the right relations [14].

Figure 1.2: Different types of causal relations.

spatial cluster of these data can represent a meaningful concept, the causal relationships between these clusters over the time series indicate the mechanism of operation. Thus, the objective of this thesis is to find two spatial clusters of the data that have a causal relationship with each other, as shown in Fig 1.2a. This is in contrast to a standard causal relation graph from random variables, where a causal relation, or relations graph, is extracted from a set of random variables with no regard to time of occurrence[14], shown in Fig 1.2b.

There were many existing works for identifying causal relationships within purely temporal or spatial data. Granger causality [15], for example, can identify causal relations between time series, or PC algorithm [16] for discrete random variables. When extended to spatio-temporal data, even though several works can identify non-causal relations [1], none can identify causal relations. With causal relations, a more thorough understanding of the occurrence mechanism could be achieved.

A brief comparison of existing works is shown in Table 1.1. This table shows the

Table 1.1: A brief comparison of different spatio-temporal relation mining methods

Field/Method	**Input**	**Output**	
		Spatial division	**Type of relations**
Geoinformatics [17]	Multivariate time-series	Predefined	**Causality**
Neuroscience [18]	Spatio-temporal raster	**Estimated**	**Causality**
Cluster Sequence Mining [20]	**Spatio-temporal Point-process**	**Estimated**	Co-occurrence
Proposed Method	**Spatio-temporal Point-process**	**Estimated**	**Causality**

difference in relationship mining within spatio-temporal data. In the geoinformatics field [17], the spatial part is just a sensor located at different locations, thus making the data a multivariate time series. They then try to find causality from these data, with the aforementioned sensor locations forming a predefined spatial division. On the other hand, in neuroscience [18], they usually work with a *spatio-temporal raster*, a series of images. This work aimed to find causality between regions from these data.

Recently, Co-occurrence Cluster Mining (CCM) [19] and Cluster Sequence Mining (CSM) [20] algorithms were previously proposed algorithms directly for extracting relationships between spatial clusters from the point-process spatio-temporal data, namely, a co-occurrence relationship. The CSM algorithm can successfully find the correlation between earthquake occurrences during the 2011 Great East Japan earthquake, as shown in Fig. 1.3. However, since correlation does not imply causation, the result from those algorithms cannot be considered causal relations.

The concrete problem of this work is detailed in Fig. 1.4. With spatio-temporal point process data (Fig. 1.4a), this work aims to, firstly, perform spatial clustering of the point process (Fig. 1.4b), before trying to find pairs of spatial clusters that have a causal relationship with each other (Fig. 1.4c and 1.4d).

To solve the aforementioned problem, in this thesis, the Granger Cluster Sequence Mining (G-CSM) algorithm is proposed. It is an extension of the Cluster Sequence Mining (CSM) algorithm. The Granger Causality [15] method for causality inference was integrated. Granger causality is one of the most commonly used temporal causality analysis techniques. It originated from the field of economics, where it is being used to analyze the relationships between different time series. The principle of Granger causality is that if **A** causes **B**, then **B** must be easier to predict using all available

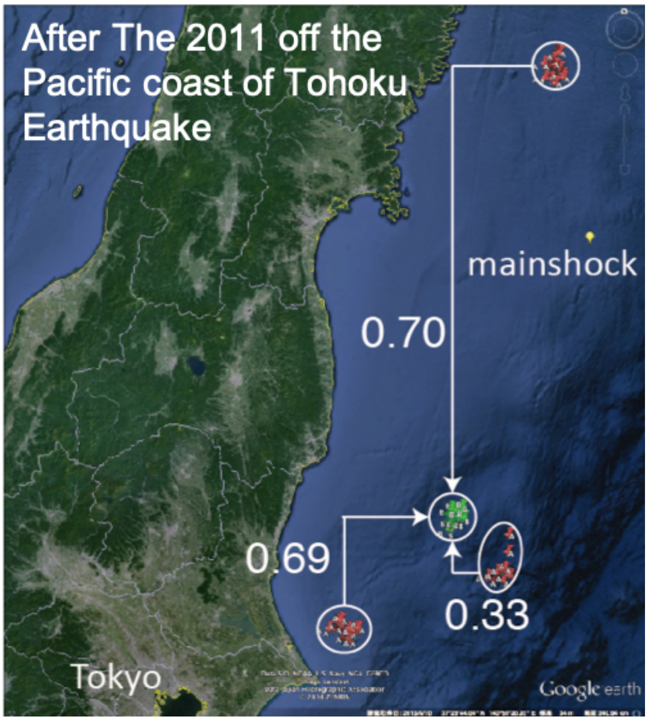
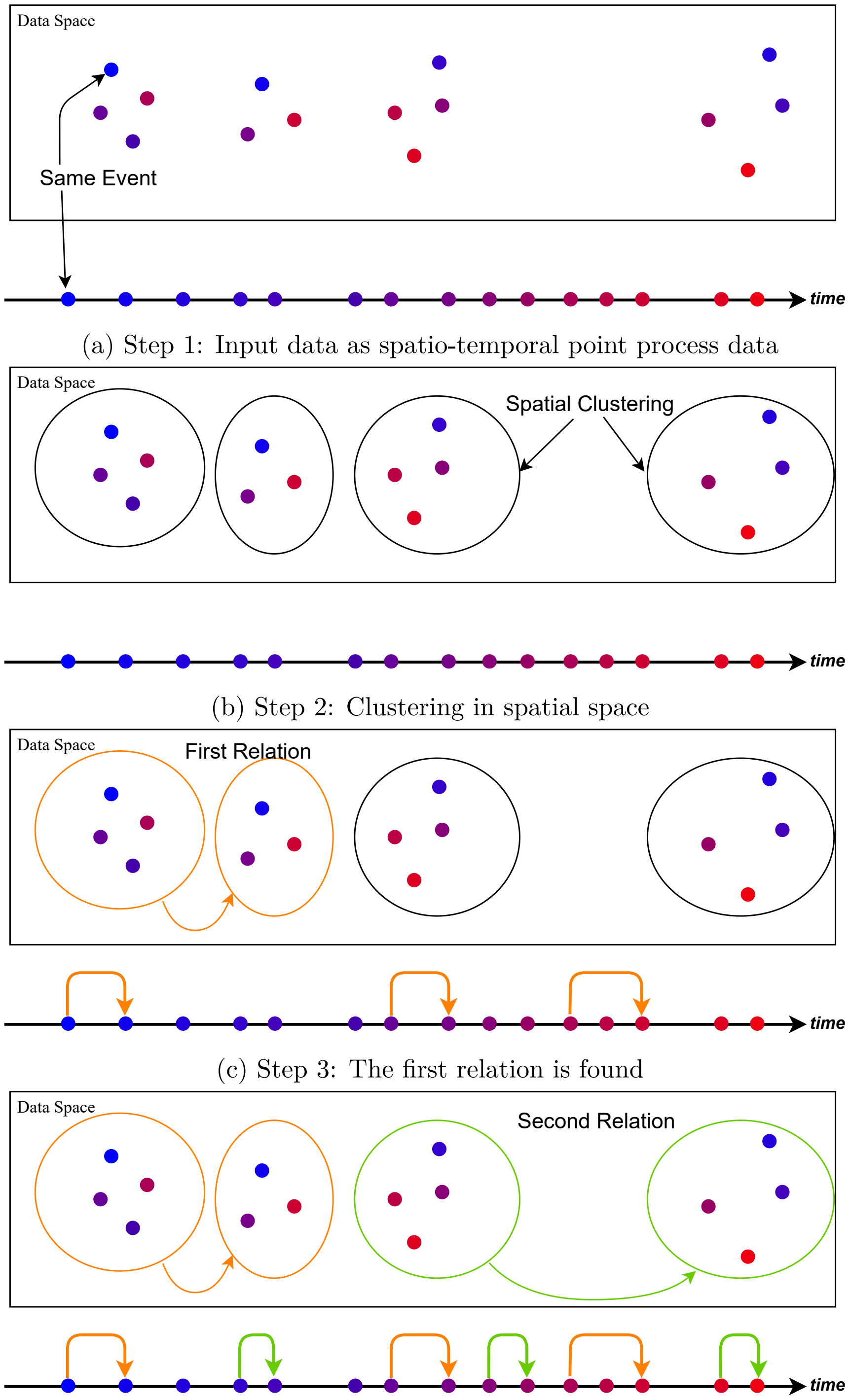


Figure 1.3: Correlation between spatial clusters of earthquake occurrence as extracted by Cluster Sequence Mining algorithm.

data than to predict using all available data except **A**. A False Discovery Rate (FDR) method was also used to quantify the significance of the detected causality, allowing us to be certain of the statistical significance and to eliminate false positive results. The proposed algorithm can extract causal relations between spatial clusters within a point process spatio-temporal data according to the causality proposed by Granger.

The performance of our proposed G-CSM algorithm was validated against the original CSM algorithm using synthetic data. The result showed that the proposed G-CSM algorithm can detect causal relations more accurately and is more robust against noise. The hyper-parameters of the G-CSM algorithm were analyzed, and was found that the G-CSM algorithm is less sensitive to them, unlike the original CSM which required a careful setting of its hyper-parameters. The usage of FDR for statistical testing also increases the accuracy of the algorithm. The G-CSM algorithm was also applied to the semi-real world data, namely, existing real-world spatial data were used with the synthetic temporal relationships.

Furthermore, a new procedure called *local density estimation* was proposed, which is a pre-processing step to modeling the vector autoregressive (VAR) model. A VAR model is used for modeling time-series data by modeling the next step in the series



(a) Step 1: Input data as spatio-temporal point process data

(b) Step 2: Clustering in spatial space

(c) Step 3: The first relation is found

(d) Step 4: Another relation is found

Figure 1.4: Concrete problem definition

using the history of itself. The G-CSM algorithm utilized a VAR model during its causality inference steps, by dividing the timeline into multiple small time windows and creating a time series out of the presence of events in each window. By applying a kernel-density estimation as a pre-processing step to the VAR model that was used for Granger causality estimation, the procedure allowed the VAR model to better capture the precise timestamp of each data in the point process, especially on sparse data, as well as allow easy scaling to longer temporal history length by having a few parameters covering long time span, while keeping the number of inputs to the model at a manageable level.

Using a linear and Gaussian kernel density model, experiments with synthetic data generated with the Poisson model were performed. The result showed that the local density estimation procedure improved the accuracy of prediction while still maintaining the same number of inputs.

The contributions of this thesis are:

- Granger Cluster Sequence Mining (G-CSM) algorithm is proposed. The G-CSM algorithm is an extension to the existing Cluster Sequence Mining algorithm by adding Granger causality inference.

- The application of the FDR procedure for evaluation of the significance of the causality is introduced.

- Local density estimation procedure is proposed. This is a pre-processing step for vector autoregressive modeling of point process data to enhance the accuracy and robustness of the model especially on sparse data.

This thesis is organized as follows: In Chapter 2, the related literature and surveys are discussed. In Chapter 3, the Granger Cluster Sequence Mining (G-CSM) algorithm is proposed, and the local density estimation procedure is detailed in Chapter 4. The discussions are in Chapter 5, with Chapter 6 concluding the thesis.

Chapter 2

Literature review

2.1 Spatio-temporal point process

2.1.1 Intensity Function

A point process is a framework used for modeling discrete points, which are modeled as an intensity function of the existence of points over the domain — spatial domain for spatial point-process and temporal domain for temporal point process [21].

A point process can be handled in many different ways. Timestamps $T_1 < T_2 < ... < T_n$ may be considered, or the interval time $S_i = T_{i+1} - T_i$. Alternatively, a counting process may be used:

$$N(t) = \|\{T_i \leq t\}\|, \tag{2.1}$$

where $\| \cdot \|$ represent the cardinality of the set, or the interval count:

$$N(a, b] = N(b) - N(a). \tag{2.2}$$

The intensity function is modeled as the cumulative incidence function (CIF), generally in the form:

$$\lambda(t) = \lim_{\Delta \to 0} \frac{\Pr(N(t + \Delta, t] = 1)}{\Delta}. \tag{2.3}$$

The intensity function may be in the form of simple Poisson distributions or more complex distributions [22]. The spatio-temporal point process is a type of point process for use with spatio-temporal data rather than just pure temporal or spatial data.

2.1.2 First-order Separability

For the spatio-temporal point process, the intensity function can be roughly separated into two groups: with or without first-order spatio-temporal separability [8]. This distinction is based on whether the intensity function can be factored into two parts: the temporal part and the spatial part.

The models with first-order spatio-temporal separability are usually simpler and have previously been used to model and predict real-world phenomena, such as earthquakes [23, 24], which model the intensity function directly after Epidemic Type Aftershock-Sequence (ETAS) model. Other earthquake prediction models such as [9] use the Hawkes model for temporal modeling and just a kernel function for spatial modeling. There is also the work [12] which uses Neural Ordinary Differential Equations (Neural ODE) and Continuous Normalizing Flows (CNF) to model the spatial and temporal intensity function.

On the other hand, this separability usually hinders the accuracy of the model, so there were also many types of research with non-separatable intensity functions. A Marked Recurrent Temporal Point Process model [25] extracted the spatial part into a feature vector and used these features as a part of their temporal model. Some use deep learning to model the intensity function [26] by creating representative points in the spatio-temporal space and calculating the final intensity as a function of these representative points.

In general, the spatio-temporal point process framework deals with modeling the spatio-temporal discrete data using a point process framework. While that is useful to study the occurrence mechanism and predict future events, the relationships between each event in the model are not explicitly defined. Only a few works exist on the topic of extracting relations of clusters in spatio-temporal point process modeling as discussed in Section 2.2.

Higuchi et al. proposed a model [27] that uses an expectation-maximization algorithm over both the Gaussian mixtures (the spatial part) and their temporal influences on each other (the temporal part). Their model can discover latent influences between each spatial cluster, however, there are two main limitations: the number of possible spatial clusters is fixed, and the relationship is derived from the coefficients of the model predictors and not a definite causal relationship. Alternatively, Zhu et al. [28] proposed a deep learning model that can generate heatmaps of spatial influence for each spatial area for interpretation.

2.1.3 Spatial Cluster with Temporal Relation

On the other hand, several works use the Spatio-temporal point process differently. That is, there are spatial parts in the data, but those spatial parts are fixed points. The example includes neural activities modeling [29]. This work does not consider these types of data, as those are better modeled as multivariate time series.

2.2 Relation mining

Relation mining is a type of data mining where a relation between each random variable is needed to be determined. This can be many kinds of relationships: similarity, dissimilarity, causal, or co-occurrence, for example. Specifically for spatio-temporal data, relationships can be defined in many ways, for example, the similarity between occurrence patterns between two spatial areas. However, this work mainly considers the causal or co-occurrence relation between two spatial areas. Even under this definition, the relation can still be divided into two types: where cause-effect occurs at the time same (no time lag), and where the effect occurs after cause (with time lag).

In the case where no time lag is observed between the associating entities, there are many types of research, especially in neuroscience. Davidson et al. proposed a network discovery algorithm using constraint tensor analysis from fMRI data [18], in which the proposed algorithm can identify the node and relation between each brain region. An example is shown in Fig. 2.1.

However, when a time lag element is added, this field becomes nearly non-existent. Methods proposed for determining time-lagged relationship included a Co-Occurrence Sequence Mining (CCM) [19] and Cluster Sequence Mining (CSM) [20].

CCM algorithm is an algorithm designed to extract a non-directional occurrence correlation from the spatio-temporal event sequence. It worked by trying to first cluster the data spatially, and then evaluate the co-occurrence coefficient of each pair of clusters. CSM extends the CCM algorithm by adding a directional requirement and using a probability inference of the lagged time (called time interval). However, both CCM and CSM algorithms were for occurrence correlation and not causal relations.

Frequent Pattern Mining might also be related to relation mining. Frequent pattern

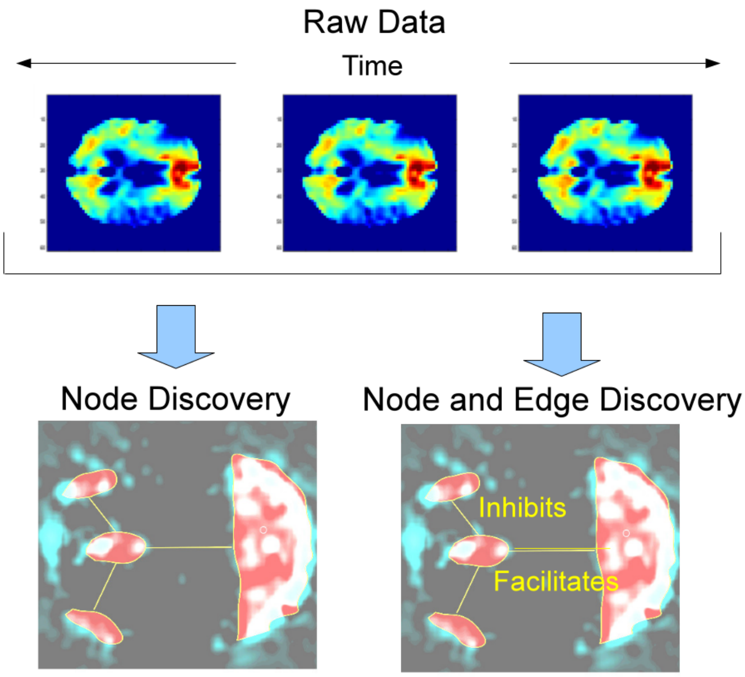


Figure 2.1: Network relationship discovery from fMRI data. Figure is taken from [18]

mining is a technique to detect patterns that occur frequently in the data. For spatio-temporal data, a variation of frequent pattern is defined including co-occurrence patterns [30], sequential pattern [31], motif pattern [32], and network pattern [33].

2.3 Causal Inference

Causal inference is a process of concluding that there exists a causal relation based on observed data. In science, causal inference is usually performed with a statistical method using techniques that will be discussed below. Nevertheless, there are many challenges in this field, mainly that *correlation does not imply causation*.

However, the word "causality" is not very well-defined. Granger was the first person to define the word causality back in 1969 [15]. Granger stated that *A causes B if it is easier to predict B using all available data than to predict B with all available data using A*. Granger's definition has been used extensively in economics. On the other hand, Spirtes et al. introduced that there is no real definition of causality [34]. Instead, all algorithms or techniques for causality inference must be verified using the "truth", be it either data that causalities are known, or synthetic data.

Exploratory causality analysis (ECA) is a practice of inferring causal associations in observed data using a statistical algorithm [34]. Specifically, ECA states that there exists a data analysis technique that can identify random variables from the data collected during a well-designed experiment, and infer a probable causal association between them.

There are many kinds of research and techniques on causality analysis and causal inference. The techniques can be divided into two main groups: for random variables and for time series. Causality on random variables, basically, has the input of multiple instances of random variables, and trying to infer causality between each variable that is true across all instances. For causality on time series, usually, the objective is to find an effect that is caused by something earlier in the time. This also creates time constraints on the possible causality, as it cannot travel back in time.

The following popular techniques will be briefly discussed:

1. *Time series:*

    - Granger causality
    - Transfer entropy

2. *Random variables:*

    - Constraint-based algorithms

Note that there were also other methods of causality inference other than explained in this section, but most of the other methods are extensions or applications of the discussed methods.

2.3.1 Granger Causality

Granger causality was among the earliest and most accepted methods for causality analysis in temporal data. As stated above, the Granger causality technique is based on the idea that A causes B if it is easier to predict B using all available data, than to predict B with all available data using A.

The standard form of Granger causality was to use a vector autoregressive (VAR)

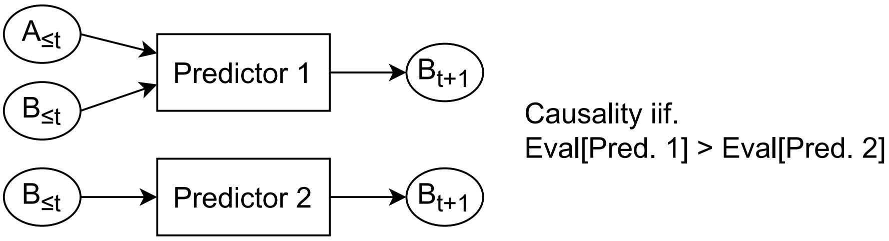


Figure 2.2: A simplified view of Granger Causality model

model for pairwise causality analysis as follows.

$$A(t) = \sum_{j=1}^{p} \Theta_{aa,j} A(t-j) + \sum_{j=1}^{p} \Theta_{ab,j} B(t-j) + E_A(t)$$
$$B(t) = \sum_{j=1}^{p} \Theta_{ba,j} A(t-j) + \sum_{j=1}^{p} \Theta_{bb,j} B(t-j) + E_B(t).$$

Here, both variable A and B were the time series being predicted using the history of both variables, $E_A(t)$ and $E_B(t)$ are residual (error) terms, Θ is the model parameters, and p is the model order. The residual terms indicated the ability of the history to predict the new value. If alternate models were created without each other history:

$$A(t) = \sum_{j=1}^{p} \Theta'_{aa,j} A(t-j) + E'_A(t)$$
$$B(t) = \sum_{j=1}^{p} \Theta'_{bb,j} B(t-j) + E'_B(t),$$

then this model represented *predicting a variable using all available information except the causal variable* in the definition proposed by Granger. If $E'_A(t) > E_A(t)$, then B caused A, and similarly if $E'_B(t) > E_B(t)$, then A caused B. The simplified diagram of the process is shown in Fig. 2.2.

There was also a spectral Granger causality [35]. Spectral Granger causality uses Fourier transform to transform the input data into a spectral domain, and detect the causality on the transformed data. This type of Granger causality was being utilized mainly in neuroscience applications.

Finally, the Granger causality for the spatio-temporal point process has also been proposed [36]. This is discussed in Chapter 3.

This work uses the Granger Causality as the causality inference method because it is the most widely used method. It is also the only causality inference method that has a point-process variation, thus, this work utilized Granger causality.

2.3.2 Transfer Entropy

Transfer entropy is a statistical method to measure the amount of directional data between two random processes [37]. The transfer entropy measured the amount of uncertainty reduced in knowing the future of B if the history of A is known.

Specifically, if there are two random processes $A(t)$ and $B(t)$, transfer entropy from A to B can be written using Shannon entropy as:

$$T_{A\to B} = H\left(B(t)|B(t-1:t-L)\right) - H\left(B(t)|B(t-1:t-L), A(t-1:t-L)\right),$$

where $H(X)$ is a Shannon entropy of X, and L is the history length to consider. The Shannon entropy represents the amount of information that is available in the random processes and its history. Hence, this measures how much entropy is lost by introducing another process into the mix. However, for the Gaussian process, it has been shown that transfer entropy and Granger causality are equivalent [38].

2.3.3 Constraint-Based Algorithms

The constraint-based algorithms differ significantly from the previously discussed causality analysis algorithms. In the case of Granger causality and transfer entropy, a causal direction is known beforehand (because they have time constraints from the temporal feature). However, when the causality direction is not known as constraints, the above algorithms fail.

In constraint-based algorithms, firstly, random variable dependencies and independencies were inferred from the observed data, using Hoeffding's test of independence or other statistical testing methods. All probable causal models are created (including with hidden cause) and are tested if the model fits with the observed (in)dependencies. The only causal relations that can be inferred are those that exist in all the valid models.

While the above algorithm can be done naively, the state-of-the-art algorithm for doing the constraint-based causal analysis is the PC algorithm [34], named after the inventor, Peter and Clark. It involved a computational optimization to allow a speedup over a naive implementation. Recently, there are also more parallel version [16] and

GPU version [39]. There was also an algorithm designed to deal with non-Gaussian data called LiNGAM [40].

Chapter 3

Granger Cluster Sequence Mining (G-CSM)

3.1 Overview

The Granger Cluster Sequence Mining (G-CSM) algorithm extended the original Cluster Sequence Mining (CSM) by implementing the Granger causality inference method to detect causality. This is done by adapting the Granger casualty to work with point process data.

This chapter first describes the working of the original CSM algorithm, then the Granger Causality and its adaption to the point process data. Next, the actual G-CSM algorithm is described. The False Discovery Rate (FDR) algorithm was introduced to control the false positive rate of the causality inference. Finally, the experiments and results are discussed.

3.2 Cluster Sequence Mining (CSM)

Cluster Sequence Mining (CSM) [20], on which the Granger Cluster Sequence Mining (G-CSM) is based, is an algorithm designed to identify occurrence correlations in the multidimensional event sequence. In this section, the detail of the original CSM algorithm is described.

Definition 3.1. *A* Multidimensional Event Sequence *is a sequence of length N of n-*

dimensional vectors of real number representing events*, each with an associated* timestamp*, ordered sequentially:*

$$X = \{\mathbf{x}^{(i)} \in \mathbb{R}^n\} \quad (|X| = N) \tag{3.1}$$

$$X_t = \langle t(\mathbf{x}^{(1)}) \leq t(\mathbf{x}^{(2)}) \leq ... \leq t(\mathbf{x}^{(N)}) \rangle. \tag{3.2}$$

An example of an event sequence is shown in Fig. 1.1b with 15 events in both spatial (data space) and temporal (timeline) view.

The CSM algorithm took an input of *multi-dimensional event sequence* as defined in Def. 3.1, and produced a *cluster sequence pattern*, which is defined as follow:

Definition 3.2. *A* Cluster Sequence Pattern *is a pair of spatial clusters of the event sequence, called* prior *cluster and* posterior *cluster.*

$$S_{A \to B} = \langle \mathbf{A} = \{\mathbf{x}^{(i)} | A^i = 1\}, \mathbf{B} = \{\mathbf{x}^{(i)} | B^i = 1\} \rangle, \ (\mathbf{A} \cap \mathbf{B} = \emptyset), \tag{3.3}$$

where A and B is an assignment vector for set **A** *and* **B** *respectively. The set* **A** *is a* prior cluster*, while set* **B** *is a* posterior cluster*.*

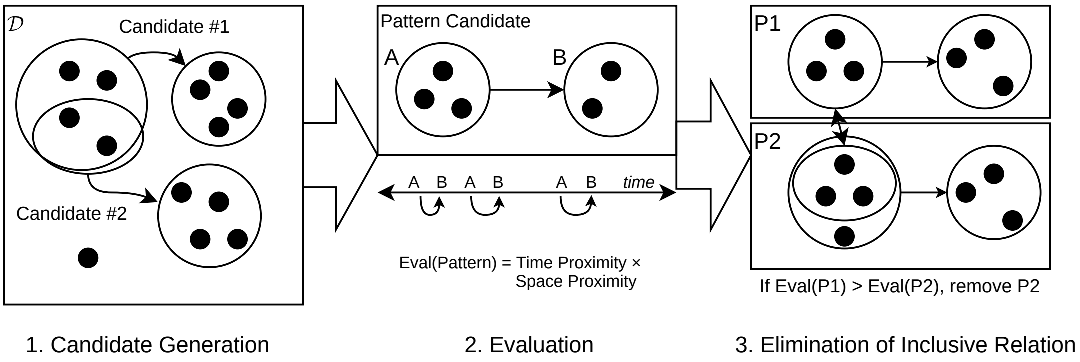
An example of a cluster sequence pattern is shown in Fig. 1.2a, created from the event sequence in Fig. 1.1b.

The objective of the CSM algorithm is to find cluster sequence patterns that satisfy the following three conditions:

1. **Temporal proximity** Each event in the posterior cluster $\mathbf{x}^{(b)} \in \mathbf{B}$ occurs immediately or soon after some event in prior cluster $\mathbf{x}^{(a)} \in \mathbf{A}$. The time interval between the two events, $t_{ab} = t(\mathbf{x}^{(b)}) - t(\mathbf{x}^{(a)})$ must be a positive number and follow some distribution $\Psi(t_{ab})$.

2. **Frequency** The more frequent $\mathbf{x}^{(a)} \in \mathbf{A}$ and $\mathbf{x}^{(b)} \in \mathbf{B}$, the better the cluster sequence. The number of pairs of events in the cluster sequence pattern must be larger than some hyper-parameter $Supp_{min}$.

3. **Spatial proximity** The variance of the event within each cluster **A** or **B** must be low. This was evaluated using the SSW (sum of squares within) measure. SSW of **A** and **B** were evaluated independently.


𝒟
Candidate #1
Candidate #2
Pattern Candidate
A
B
A B A B A B time
Eval(Pattern) = Time Proximity × Space Proximity
P1
P2
If Eval(P1) > Eval(P2), remove P2

1. Candidate Generation 2. Evaluation 3. Elimination of Inclusive Relation

Figure 3.1: Overview of CSM algorithm

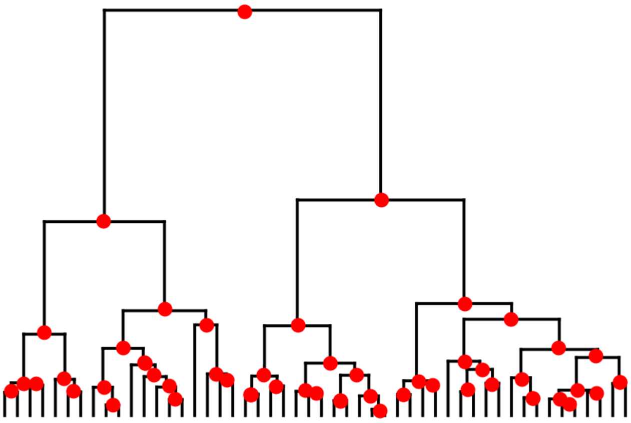


Figure 3.2: AHC dendrogram nodes

To find cluster sequence patterns that met these conditions, the CSM algorithm operated in three steps: 1) Candidate Generation 2) Candidate Evaluation 3) Elimination of Inclusive Relation. The overall process of the CSM algorithm is shown in Fig. 3.1.

Candidate Generation

For the candidate generation process, CSM used Agglomerative Hierarchical Clustering (AHC). Each node in the AHC dendrogram was a possible cluster to be chosen, as shown in Fig. 3.2.

The candidates were generated by trying to pair all possible clusters from the AHC dendrogram, and checking if they met the *frequency* requirement (requirement 2) by calculating each corresponding event. If the pair of clusters had the number of correspondent events at least $Supp_{min}$, it was considered to be a candidate cluster sequence pattern.

In this work, a simple one-to-one matching method was used to calculate the corresponding events. The simple one-to-one matching method considered each event in the prior cluster **A** separately. For each event in the prior cluster **A**, the closest event in posterior cluster **B** that had not been chosen was selected as the corresponding event. This matching algorithm is shown in Alg. 3.1.

Algorithm 3.1 CSM one-to-one event matching algorithm

```
    **Input** List of timestamps of prior and posterior $\mathbf{A}$ and $\mathbf{B}$
    **Output** List of time interval
$L_b \leftarrow 0$
$T \leftarrow \langle\rangle$
**for all** $a \in \mathbf{A}$ **do**
    **while** $L_b < |\mathbf{B}| \wedge \mathbf{B}[L_b] < a$ **do**
        $L_b \leftarrow L_b + 1$
    **end while**
    **if** $L_b < |\mathbf{B}|$ **then**
        Append $(\mathbf{B}[L_b] - a)$ to $T$
    **end if**
**end for**
**return** $T$
```

Evaluation

The pattern candidates were evaluated using the following evaluation functions:

$$\mathcal{F}(\hat{\lambda}_{AB}) = \frac{1}{1 + \exp(-\tau\hat{\lambda}_{AB})}, \tag{3.4}$$

$$\mathcal{G}(\mathbf{A}, \mathbf{B}) = \exp\left(-\frac{\mathrm{SSW}(\mathbf{A})^2 + \mathrm{SSW}(\mathbf{B})^2}{2\sigma^2}\right), \tag{3.5}$$

$$\mathcal{L}(S_{A\to B}) = \mathcal{F}(S_{A\to B})^{\gamma} \cdot \mathcal{G}(\mathbf{A}, \mathbf{B})^{(1-\gamma)}. \tag{3.6}$$

$\mathcal{F}$ is a time proximity evaluation according to the *time proximity* requirement (requirement 1). The greater the $\mathcal{F}$, the higher the time proximity. This work assumed that the lower the time interval between the events, the better the cluster sequence. Thus, $\Psi(t_{ab})$ is an exponential distribution $\mathrm{Exp}(\hat{\lambda}_{AB})$. The variable $\hat{\lambda}_{AB}$ is the maximum likelihood parameter from the observed time interval. Thus, the higher the $\hat{\lambda}_{AB}$ the better the temporal proximity. The value was then normalized using the sigmoid function to be in a range of $[0, 1]$. The hyper-parameter of the sigmoid function, τ, is used to control the relative resolution.

$\mathcal{G}$ is a space proximity function for the *space proximity* requirement. Similarly, the higher the value of this function, the higher the spatial proximity. The sum of square within (SSW) measure was used to evaluate each prior and posterior cluster separately. SSW measures the variance of the data in relation to the cluster center. The values were combined and normalized to $[0, 1]$ using a Gaussian function, with σ being a hyper-parameter to control the relative resolution.

$\mathcal{L}$ is the final evaluation score that combined both spatial and temporal proximity. The final evaluation was constructed from the product of time and proximity function, weighted by the parameter γ. The higher γ, the more important is the time proximity. An equal weight would be $\gamma = 0.5$. The higher the final evaluation score, the better the cluster sequence satisfied all 3 CSM requirements. Only the cluster sequence pattern candidates with $\mathcal{L}(S_{A\to B}) \geq \mathcal{L}_{min}$, where $\mathcal{L}_{min}$ is a predefined minimum threshold, were considered as the final cluster sequence patterns.

Elimination of Inclusive Relation

If any two final cluster sequence patterns have an inclusive relation with each other, then only the pattern with a higher evaluation score was kept.

Any two patterns were considered to have an inclusive relation with each other when the prior cluster of one relation is a subset of the prior cluster of the other relation, and the posterior cluster is also a subset of the other. Note that the subset may be in a different direction for the prior and posterior clusters.

$$\text{ClusterInclusive}(\mathbf{X}, \mathbf{Y}) = \begin{cases} 1 & \mathbf{X} \subseteq \mathbf{Y} \\ 1 & \mathbf{Y} \subseteq \mathbf{X} \\ 0 & \textit{otherwise} \end{cases} \tag{3.7}$$

$$\text{PatternInclusive}(S_{A\to B}, S_{C\to D}) = \text{ClusterInclusive}(\mathbf{A}, \mathbf{C}) \cdot \text{ClusterInclusive}(\mathbf{B}, \mathbf{D}) \tag{3.8}$$

This operation can be performed efficiently by checking the AHC dendrogram. If any two clusters are (grand)parent/(grand)child of each other, the two clusters have an inclusive relation.

3.3 Granger Causality

Granger causality [15] is causality testing based on the idea that: if A causes B, then it must be easier to predict B using all available data than to predict B using all available data except A.

Assuming a data A with $A(t)$ representing the data A from time 0 to time t. Let $P(A(t)|X(t))$ be an optimum predictor of $A(t)$ based on data $X(t)$, and let $U(t)$ be all available data up to time t. Denote the predictive error of $P(A(t)|X(t))$ with $\varepsilon(A(t)|X(t))$, and the variance of $\varepsilon(A(t)|X(t))$ as $\sigma^2(A(t)|X(t))$.

Definition 3.3. Granger Causality. *If $\sigma^2(B(t+1)|U(t)) < \sigma^2(B(t+1)|U(t) - A(t))$, then A is causing B. This is written as A* granger-cause *(g-cause) B.*

In reality, it is not practical to construct a model using all the available data. Generally, the data used to model the predictor is usually limited to the observed variables. For pairwise causality, only two random variables are considered.

In addition to the number of variables, there is also the problem of history length. Using all available data, even just for two random variables, means using all observations since the beginning. This is not practical for many reasons. The standard practice is to limit the history to some small time frame.

Pairwise Point Process Granger Causality

Traditionally, the Granger causality works on time series or spectral data. However, in this work, such data was not available. The only temporal data available was the point process of the timestamps of the event occurrence time. A generic point process Granger causality was proposed in [36]. This sub-section contains a slight adaption of the method from the aforementioned work to fit the needs of the G-CSM algorithm.

Basically, a Granger causality of the cluster sequence pattern $S_{A \to B}$ is whether **A** g-causes **B** or not. A cumulative incidence function (CIF) for the point process of event

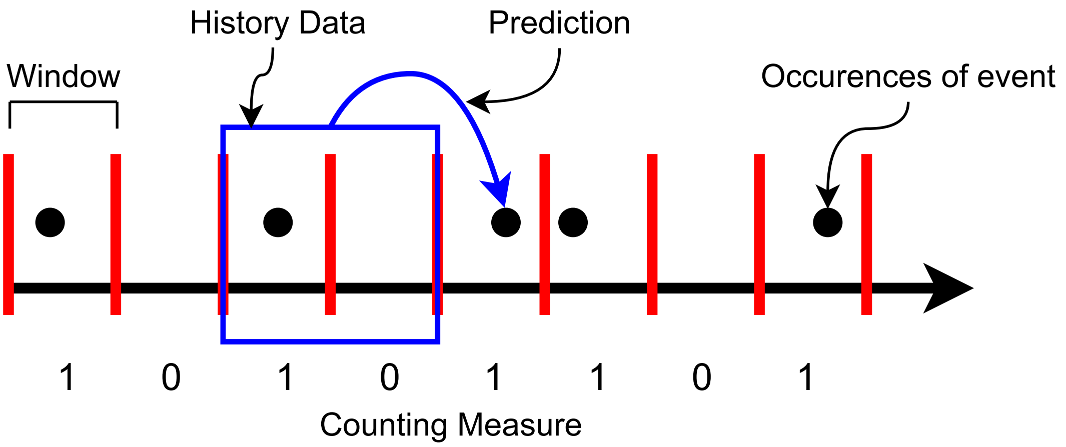


Figure 3.3: Simplified view of the GLM model for point-process Granger causality.

B occurrence can be defined as:

$$\lambda_b(t|H_b(t)) = \lim_{\Delta \to 0} \frac{Pr[N_b(t+\Delta) - N_b(t) = 1]}{\Delta}, \tag{3.9}$$

where $N_b(t)$ is a counting measure of event b within the time of $(0, t]$, $H_b(t)$ is an occurrence history of all event occurrences up to time t for event b. The probability that event b occurring in a small time window $[t, t + \Delta)$ can be written as $\lambda_b(t|H_b(t))\Delta$.

Since it is not feasible to consider the entire history, only the history from time $[t - M_i W, t]$ is considered. The time range was divided into M_i equal windows of length W. The number of occurrence of event q in the time window $[t - mW, t - (m - 1)W]$ is denoted as $R_{q,m}(t)$ for $q \in \{\mathbf{A}, \mathbf{B}\}$ is either a prior or posterior event, and $m = 1, ..., M_i$ is the window number.

To model the predictor for the event occurrence, a generalized linear model (GLM) framework was used to model the CIF. In GLM, the logarithm of the CIF was modeled using a linear combination of the occurrence history. This work currently assumes the linearity of the data by setting a small window size. The simplified model is shown in Fig. 3.3. In this case, the log-CIF is modeled as:

$$\log \lambda_b(t|\theta_b, H_b(t)) = \theta_{b,0} + \sum_{q \in \{\mathbf{A},\mathbf{B}\}} \sum_{m=1}^{M_b} \theta_{b,q,m} R_{q,m}(t), \tag{3.10}$$

where $\theta_{b,0}$ is background activity, and $\theta_{b,q,m}$ is the effect of $R_{q,m}(t)$ to the event b. The parameter vector, θ_b, is defined as:

$$\theta_b = \{\theta_{b,0}, \theta_{b,a,1}, ..., \theta_{b,b,1}, ..., \theta_{b,b,M_b}\} \tag{3.11}$$

A point process likelihood function [29] was used to fit the GLM model. As [29] shown that for the point process, both the Binomial and Poisson estimation in GLM are equivalent, this work chose the former.

To make the calculation easier, the entire timeline $[0, T]$ was divided into K equal non-overlapping windows, each with the length of W. The time window k would represent the time window $(t_{k-1} = (k-1)W, t_k = kW]$. To represent this discretized time, the history $H_b(t)$ is written as $H_b[k]$, and $R_{q,m}(t)$ is written as $R_{q,m}[k]$. $\Delta N_b[k] = N_b[k] - N_b[k-1]$ is the number of event occurrence within the time $(t_{k-1}, t_k]$, and the CIF in eq. (3.9) is written as $\lambda_b(t_k|\theta_b, H_b[k])$. W should be chosen to be very small so that $\Delta N_b[k]$ can only be either 0 or 1.

Thus, the likelihood function for the Binomial GLM model is given as:

$$L_b(\theta_b) = \prod_{k=1}^{K} \left[\lambda_b(t|\theta_b, H_b[k])\Delta\right]^{\Delta N_b[k]} \left[1 - \lambda_b(t|\theta_b, H_b[k])\Delta\right]^{1-\Delta N_b[k]}. \tag{3.12}$$

As per the Granger causality defined in Def. 3.3, event **A** is considered to granger-cause event **B** if there was a reduction in the likelihood between predicting the occurrence using history of only **B** instead of using the history of both **A** and **B**. The log-likelihood ratio, $\Gamma(S_{A\to B})$ is defined as:

$$\Gamma(S_{A\to B}) = \log \frac{L_b(\theta_b^a)}{L_b(\theta_b)}, \tag{3.13}$$

where the likelihood $L_b(\theta_b)$ was obtained from model fitting eq. (3.10), and the likelihood $L_b(\theta_b^a)$ was obtained using new CIF with history of **A** cut:

$$\log \lambda_b^a(t|\theta_b^a, H_b^a(t)) = \theta_{b,0}^a + \sum_{m=1}^{M_b} \theta_{b,b,m}^a R_{b,m}(t). \tag{3.14}$$

The log-likelihood ratio $\Gamma(S_{A\to B})$ in eq. (3.13) is considered to be a Granger causality strength for pattern $S_{A\to B}$.

3.3.1 Significance testing

The Granger causality strength from eq. (3.13) cannot tell us whether the relation is significant enough to be considered a causality. Thus, the following null and alternative hypotheses were formed:

$$H_0 : \theta' = \theta_b^a \quad \text{(the limited predictor is better)}, \tag{3.15}$$

$$H_1 : \theta' = \theta_b \quad \text{(the full predictor is better)}. \tag{3.16}$$

To test H_0 against H_1, likelihood-ratio test [41, 36] were used. The likelihood-ratio test evaluated the difference of deviance ΔD flowing the χ^2 distribution, which is given by:

$$\Delta D = -2\Gamma(S_{A\to B}) \sim \chi_w^2, \tag{3.17}$$

where w is the degree of freedom, in this case, the difference in the dimensionality of the two predictors, which is equal to the history length of Granger causality M_b.

Because the algorithm performed tens of thousands of significant tests throughout the algorithm, there is a need to control the type-I error rate. False Discovery Rate (FDR) procedure [42] was used, specifically the Benjamini–Hochberg procedure, which can be summed up as:

1. Perform all significant testings and calculate the p-values.

2. Rank all p-values from low to high. So that $p_1 \leq p_2 \leq p_3 \leq ... \leq p_n$

3. Find maximum k such that $p_k \leq \frac{k}{n}\alpha$, where α is the acceptable ratio of type-I error.

4. Accept first k testings.

In the algorithm, FDR was applied over all cluster sequence pattern candidates. The p-value of each candidate was calculated according to eq. (3.17). Threshold Γ_0 for the Granger causality strength was calculated such that:

$$P(\Gamma(S_{A\to B}) \geq \Gamma_0) = p_k, \tag{3.18}$$

where p_k is the highest p-value accepted by the FDR algorithm above, meaning Γ_0 is the likelihood ratio of the k$^{\text{th}}$ candidate pattern, sorted by the likelihood ratio.

3.4 Granger Cluster Sequence Mining (G-CSM)

Granger Cluster Sequence Mining (G-CSM) algorithm modified the original CSM as described in Section 3.2 with a Granger causality-based time proximity.

3.4.1 Time Proximity Evaluation

The temporal evaluation is based on the strength and significance of the Granger causality of the sequence. This thesis proposed two different methods of temporal evaluation: threshold strength and scaled strength.

1. **Threshold strength**. Use the significant threshold as a cutoff, resulting in the sequence that is deemed to have significant causality to be evaluated using spatial features only.

$$\mathcal{F}_{TH}(S_{A\to B}) = \begin{cases} 0 & (\Gamma(S_{A\to B}) < \Gamma_0) \\ \\ 1 & (\Gamma(S_{A\to B}) \geq \Gamma_0) \end{cases}. \tag{3.19}$$

2. **Scaled strength**. The strength is scaled from 0 to 1, with anything at or below the significant threshold yielding 0, and increasing toward 1 as the causality strength goes toward infinity.

$$\mathcal{F}_{SC}(S_{A\to B}) = \max\left(0, 1 + \frac{\Gamma_0}{\Gamma(S_{A\to B})}\right). \tag{3.20}$$

The temporal evaluation is combined with the spatial evaluation resulting in the final evaluation for G-CSM:

$$\mathcal{F}(S_{A\to B}) = \mathcal{F}_{\text{TH or SC}}, \tag{3.21}$$

$$\mathcal{G}(\mathbf{A},\mathbf{B}) = \exp\left(-\frac{\text{SSW}(\mathbf{A})^2 + \text{SSW}(\mathbf{B})^2}{2\sigma^2}\right), \tag{3.22}$$

$$\mathcal{L}(S_{A\to B}) = \mathcal{F}(S_{A\to B})^{\gamma} \cdot \mathcal{G}(\mathbf{A},\mathbf{B})^{(1-\gamma)}. \tag{3.23}$$

The spatial proximity $\mathcal{G}(\mathbf{A},\mathbf{B})$ in eq. (3.22) and the overall evaluation $\mathcal{L}(S_{A\to B})$ in eq. (3.23) is identical to the original CSM in eq. (3.5) and (3.6).

A cluster sequence pattern with Granger causality is called a *Granger cluster sequence pattern* (*g-pattern*). The full pseudocode of the G-CSM algorithm is shown in Alg. 3.2.

3.5 Experiments

To validate the proposed algorithm, multiple experiments were performed with synthetic data and semi-real data. First, the performance between the proposed G-CSM algorithm with and without FDR, and the original CSM algorithm was compared. Various patterns of the synthetic data were also tested. Second, an analysis of the hyper-parameters of the proposed algorithm was performed. Lastly, a semi-real data was tested. No other method can be compared other than CSM as discussed in Section 2.3.1.

3.5.1 Data generation

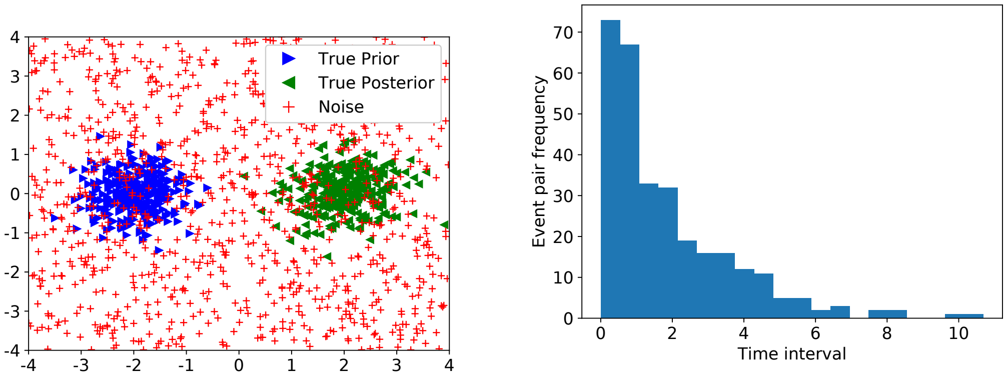
In many of the experiments, synthetic data with embedded true relations and noise were used. The synthetic data contain an *embedded relation*, which is a pair of spatial clusters that has the time interval between the corresponding event in prior and posterior event clusters following an exponential distribution. The synthetic data also have noise added. This process is similar to the synthetic data used in the CSM paper [20].

The embedded relation is generated as:

1. Generate N data points from a normal distribution for two clusters: $\mathbf{x^{(i)}} \in X \sim \mathcal{N}(m_A, \Sigma_A)$ and $\mathbf{y^{(i)}} \in Y \sim \mathcal{N}(m_B, \Sigma_B)$.

Algorithm 3.2 G-CSM algorithm

```
Input List of event $X$ and timestamps $X_t$
Output List of cluster sequence patterns
# Step 1: Candidate Generation
Perform AHC on $X$
$C \leftarrow \emptyset$                                  ▷ Set of candidates
for all $A \in \mathrm{AHC}(X)$ do
    for all $B \in \mathrm{AHC}(X)$ do
        if $A = B$ then
            Continue.
        end if
        $T \leftarrow \min(\|A\|, \|B\|)$
        if $T \geq Supp_{min}$ then
            Append $\langle A, B \rangle$ to $C$
        end if
    end for
end for
# Step 2: Evaluation
$D \leftarrow \emptyset$                  ▷ Set of preliminary sequence patterns
for $S_{A \rightarrow B} \in C$ do
    $L = \mathcal{L}(S_{A \rightarrow B})$                          ▷ eq. (3.23)
    if $L \geq \mathcal{L}_{min}$ then
        Append $S_{A \rightarrow B}$ to $D$
    end if
end for
# Step 3: Elimination of Inclusive Relation
Sort $D$ by evaluation score, high to low.
for $i$ from 1 to $\|D\|$ do
    for $j$ from $i + 1$ to $\|D\|$ do
        if PatternInclusive(D[i], D[j]) then
            Remove $D[j]$ from $D$
        end if
    end for
end for
return $D$
```



(a) Spatial view of data with $N_{noise} = 1000$

(b) Histogram of the intervals between relation, $\lambda = 0.5$

Figure 3.4: Example of the input data

2. For each pair of $\langle \mathbf{x^{(i)}}, \mathbf{y^{(i)}} \rangle$, generate a $t^{(i)} \sim \text{Exp}(\lambda)$.

3. Set $t_{Gap} = ((t_1 - t_0) - \sum t^{(i)})/(N-1)$, the gap between each event. The input parameter t_0, t_1, and λ should be set such that $t^{(i)} \ll t_{Gap}$.

4. Each pair of $\langle \mathbf{x^{(i)}}, \mathbf{y^{(i)}} \rangle$ are allocated a timestamp such that $t(\mathbf{x^{(i)}}) = t_0 + (i-1)t_{Gap} + \sum_{j=0}^{i-1} t^{(j)}$ and $t(\mathbf{y^{(i)}}) = t_0 + (i-1)t_{Gap} + \sum_{j=0}^{i} t^{(j)}$.

The noise is generated as a uniform spatial noise uniformly over the timeline:

1. Generate N data points from a uniform distribution: $\mathbf{x^{(i)}} \sim \mathcal{U}[\mathbf{a}, \mathbf{b}]$.

2. For an event $\mathbf{x^{(i)}}$, set $t(\mathbf{x^{(i)}}) \sim \mathcal{U}[t_0, t_1]$

In this experiment, the embedded relation was generated using the parameters $N = 300$, $m_A = (-2, 0)$, $\Sigma_A = (0.5, 0.5)$, $m_B = (2, 0)$, $\Sigma_B = (0.5, 0.5)$, $t_0 = 0$, and $t_1 = 100,000$. Noises were generated using the parameters: $\mathbf{a} = (-4, -4)$, $\mathbf{b} = (4, 4)$, $t_0 = 0$, $t_1 = 100,000$. This created a single cluster sequence pattern, with noise directly over the event cluster, to test the basic accuracy of the algorithm. The number of noise, N_{noise} is varied by each experiment.

The example spatial view of the data with $N_{noise} = 1000$ and the histogram of the interval between each relation with $\lambda = 2$ (average interval $= 0.5$) are shown in Fig. 3.4.

3.5.2 Evaluation measure

The precision, recall, and F-score of the identified prior and posterior clusters, and of the relation itself, were measured. The equations for these scores are as follows:

$$Prec(C) = \frac{\|C \cap X\|}{\|C\|}, \tag{3.24}$$

$$Rec(C) = \frac{\|C \cap X\|}{N}, \tag{3.25}$$

where $Prec(C)$ and $Rec(C)$ are the precision and recall score of the cluster C given the ground-truth cluster X. $\| \cdot \|$ denoted the cardinal of the event set. F-scores were calculated as the harmonic mean between the precision and recall score.

This work defined relation-based precision and recall as follows:

$$Prec(S_{A \to B}) = \frac{\|\{i | \mathbf{x}^{(i)} \in A \wedge \mathbf{y}^{(i)} \in B\}\|}{0.5 \times (\|A\| + \|B\|)}, \tag{3.26}$$

$$Rec(S_{A \to B}) = \frac{\|\{i | \mathbf{x}^{(i)} \in A \wedge \mathbf{y}^{(i)} \in B\}\|}{N}, \tag{3.27}$$

where $\mathbf{x}^{(i)} \in X$ and $\mathbf{y}^{(i)} \in Y$ are ground-truth prior and posterior event clusters, with $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ being the associated event pair as generated in Section 3.5.1, and N is the number of event pairs in the ground-truth relations. The precision and recall score of each relation was calculated using the number of pairs of events that were actually related to each other in that relation.

Outputted relations were also counted. The number of relations in the generated data is one. However, the algorithm may output more than one relation, whether because it detected more than one relation in the input, or because multiple subsets of the same relation were detected. In such cases, the prior clusters of all relations were merged, and the posterior clusters were also similarly merged for the purpose of evaluation only.

3.5.3 Performance validation

This section compared the accuracy of the original CSM algorithm with the proposed G-CSM algorithm with and without FDR.

Table 3.1: AIC value at different history length. Bold is the lowest.

	History				
Noise	2	3	5	8	15
100	2928	2856	2781	**2760**	2771
500	3333	3269	3206	**3191**	3202
1000	3722	3675	3632	**3625**	3637
2000	4644	4621	**4603**	**4603**	4616
3000	5344	5333	**5326**	5330	5344

Parameter settings

The hyper-parameters were set with $\sigma = 0.5$, $\gamma = 0.5$, $\mathcal{L}_{min} = 0.8$. These parameters were set to balance the effect of the spatial and temporal scores.

For the Granger causality, the window size $W = 1$ was used. The history length, M_b was set using Akaike Information Criteria (AIC) [43] on the Granger causality model according to Table 3.1. α for significant testing is set to 0.05, which means 5% error rate for causality detection was accepted.

The $Supp_{min}$ parameter, which controls the number of corresponding events in each pattern candidate, was set to 50. This meant that all candidate patterns must have at least 50 events within each cluster.

Result with varying noise level

Firstly, the algorithms were tested with various amounts of noise (N_{noise}) and $\lambda = 0.5$. The G-CSM using $\mathcal{F}_{TH}$ (eq. (3.19)) is denoted with **FDR-TH**, while the one using $\mathcal{F}_{SC}$ (eq. (3.20)) is denoted with **FDR-SC**. The result is shown in Table 3.2. All results were an average of 20 runs.

The example of relations extracted by each algorithm is shown in Fig. 3.5. The G-CSM without FDR and G-CSM with FDR-SC result are similar, but G-CSM with FDR-SC result has a slightly bigger posterior cluster, which better matches the generated data. The G-CSM with FDR-TH has a lower precision cluster and sometimes identifies an erroneous relation as shown.

Meanwhile, the original CSM failed to extract any relations at all even with noise = 100. The main reason is that the original CSM tried to match each event pair together, so by having noise without matching pairs in the spatial cluster, it failed to detect any meaningful relations. In contrast, G-CSM uses window-based event counting in the time-space, therefore G-CSM is more robust against noise in the time domain than

Table 3.2: CSM and G-CSM at various noise levels. P = Precision, R = Recall, F = F-score, (1) = Prior cluster, (2) = Posterior cluster, (R) = Relation, Cnt. = Number of relations identified. Bold indicated the best result.

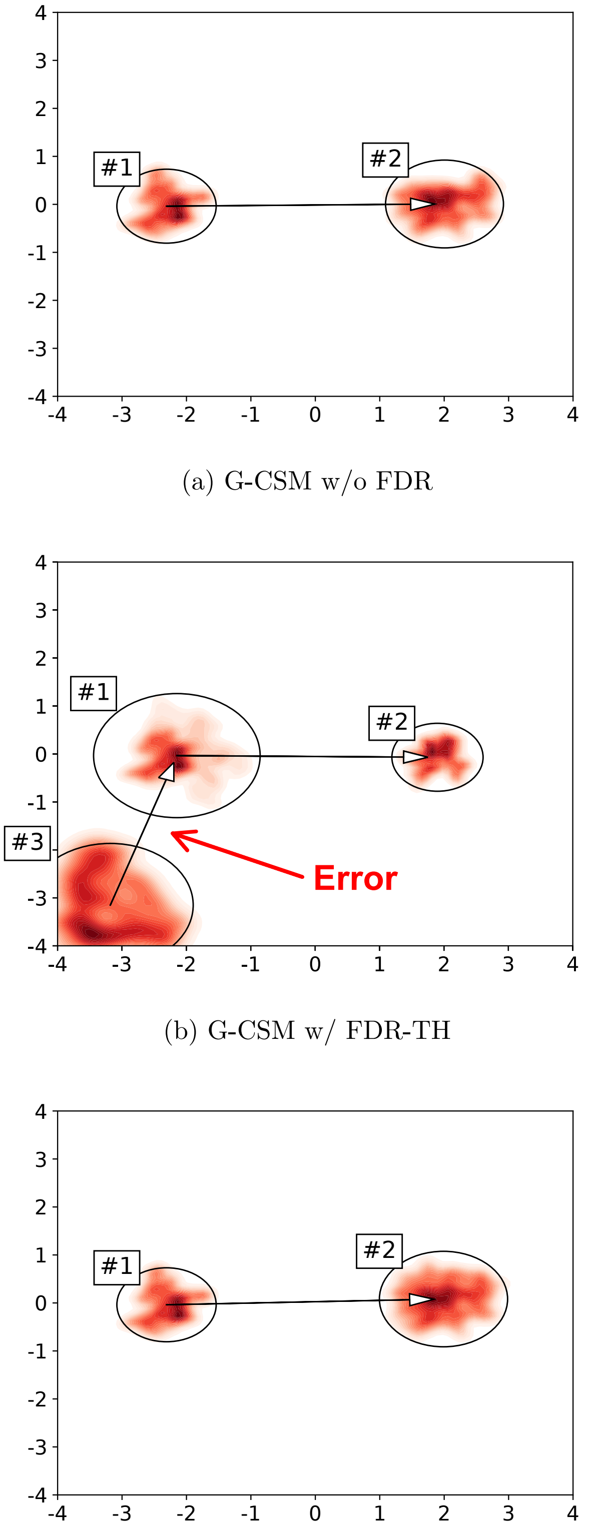
Algo.	**P (1)**	**R (1)**	**F (1)**	**P (2)**	**R (2)**	**F (2)**	**P (R)**	**R (R)**	**F (R)**	**Cnt.**
Noise = 100										
G-CSM	0.987	**0.540**	**0.693**	0.987	**0.548**	**0.701**	**0.534**	**0.297**	**0.381**	**1.050**
FDR-TH	0.987	0.355	0.512	**0.989**	0.327	0.485	0.324	0.116	0.169	1.250
FDR-SC	**0.989**	0.495	0.654	0.988	0.492	0.649	0.473	0.242	0.319	**1.050**
Noise = 500										
G-CSM	0.940	**0.556**	**0.695**	0.936	**0.537**	**0.678**	**0.510**	**0.301**	**0.377**	**1.100**
FDR-TH	0.921	0.389	0.507	0.912	0.362	0.504	0.318	0.143	0.187	1.450
FDR-SC	**0.941**	0.536	0.678	**0.937**	0.513	0.658	0.490	0.277	0.352	**1.100**
Noise = 1000										
G-CSM	**0.885**	0.551	0.674	**0.888**	0.549	0.673	0.484	0.305	0.373	**1.100**
FDR-TH	0.771	0.428	0.519	0.772	0.416	0.503	0.318	0.192	0.227	1.900
FDR-SC	0.883	**0.560**	**0.681**	0.887	**0.555**	**0.678**	**0.491**	**0.314**	**0.382**	**1.100**
Noise = 2000										
G-CSM	0.783	0.550	0.637	**0.791**	0.571	0.655	0.436	0.320	0.365	**1.300**
FDR-TH	0.662	0.476	0.516	0.675	0.473	0.522	0.309	0.245	0.257	2.450
FDR-SC	**0.784**	**0.560**	**0.643**	0.788	**0.604**	**0.679**	**0.449**	**0.341**	**0.385**	**1.300**
Noise = 3000										
G-CSM	0.679	0.595	0.623	**0.705**	0.609	0.643	0.407	0.366	0.381	1.850
FDR-TH	0.500	0.497	0.467	0.530	**0.634**	0.542	0.272	0.322	0.281	4.050
FDR-SC	**0.683**	**0.611**	**0.635**	0.701	0.631	**0.657**	**0.423**	**0.389**	**0.402**	**1.700**

CSM.

In Table 3.2, The FDR-TH algorithm is also uniformly bad, having worse results than the G-CSM without FDR and FDR-SC in almost all measurements. This is because the temporal score becomes either 0 or 1, the score relies entirely on the spatial evaluation. The spatial evaluation prefers compact clusters, thus, FDR-TH has high precision but low recall. FDR-TH also tends to detect erroneous relations, especially at a higher noise level.

G-CSM with FDR-SC has a better recall score than G-CSM without FDR, especially as the noise increases. This result shows the robustness of G-CSM with FDR-SC against spatial noise. G-CSM with FDR-SC also maintains the relational precision score better than G-CSM without FDR.

Note that for all algorithms, the number of extracted relations (Cnt. in Table 3.2) also increased along with the noise. This is because the algorithm finds multiple relations that are a subset of the ground-truth relation, resulting in more coverage of the ground-truth data. In turn, this increases the recall score.



(a) G-CSM w/o FDR

(b) G-CSM w/ FDR-TH

(c) G-CSM w/ FDR-SC

Figure 3.5: Example of relation(s) extracted by each algorithm with Noise = 3000. CSM cannot identify any relation.

Table 3.3: CSM and G-CSM at various relation intervals. P = Precision, R = Recall, F = F-score, (1) = Prior cluster, (2) = Posterior cluster, (R) = Relation, Cnt. = Number of relations identified. Bold indicated the best result.

Algo.	**P (1)**	**R (1)**	**F (1)**	**P (2)**	**R (2)**	**F (2)**	**P (R)**	**R (R)**	**F (R)**	**Cnt.**
λ = 20 (Interval = 0.05)										
G-CSM	**0.928**	0.683	0.781	**0.929**	**0.693**	**0.790**	0.630	0.471	0.538	**1.050**
FDR-TH	0.754	0.366	0.456	0.759	0.357	0.458	0.270	0.133	0.170	1.600
FDR-SC	0.926	**0.699**	**0.791**	0.928	0.676	0.777	**0.631**	**0.474**	**0.540**	**1.050**
λ = 2 (Interval = 0.5)										
G-CSM	0.940	**0.556**	**0.695**	0.936	**0.537**	**0.678**	**0.510**	**0.301**	**0.377**	**1.100**
FDR-TH	0.921	0.389	0.507	0.912	0.362	0.504	0.318	0.143	0.187	1.450
FDR-SC	**0.941**	0.536	0.678	**0.937**	0.513	0.658	0.490	0.277	0.352	**1.100**
$\lambda = 0.2$ ***(Interval = 5)***										
G-CSM	**0.928**	0.692	0.790	0.922	0.766	0.835	0.671	0.531	0.592	**1.000**
FDR-TH	0.870	0.398	0.513	0.875	0.399	0.524	0.328	0.161	0.208	1.500
FDR-SC	0.927	**0.698**	**0.793**	**0.923**	**0.767**	**0.836**	**0.674**	**0.536**	**0.596**	**1.000**

Result with varying temporal interval

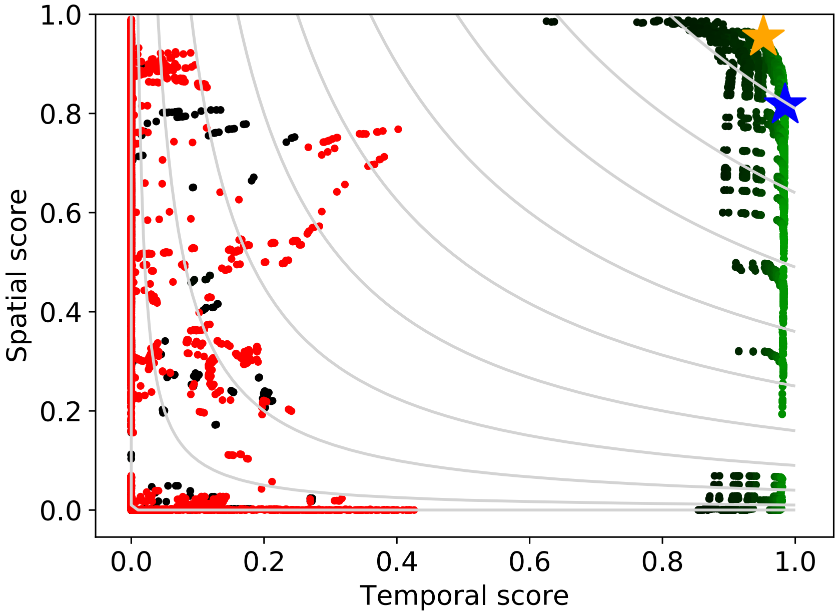
The algorithm was also tested at various settings of λ that controls the interval between each related event with noise $N_{noise} = 1000$. The result is shown in Table 3.3. All results were an average of 20 runs.

At a low interval between each pair of events, it is also a toss-up between G-CSM without FDR and G-CSM with FDR-SC. However, at a larger time interval, FDR-SC performs better. It is suspected that this is because as intervals get longer, there are more chances for the noise to appear in between the event pair in the temporal domain, and FDR-SC is more robust to noise.

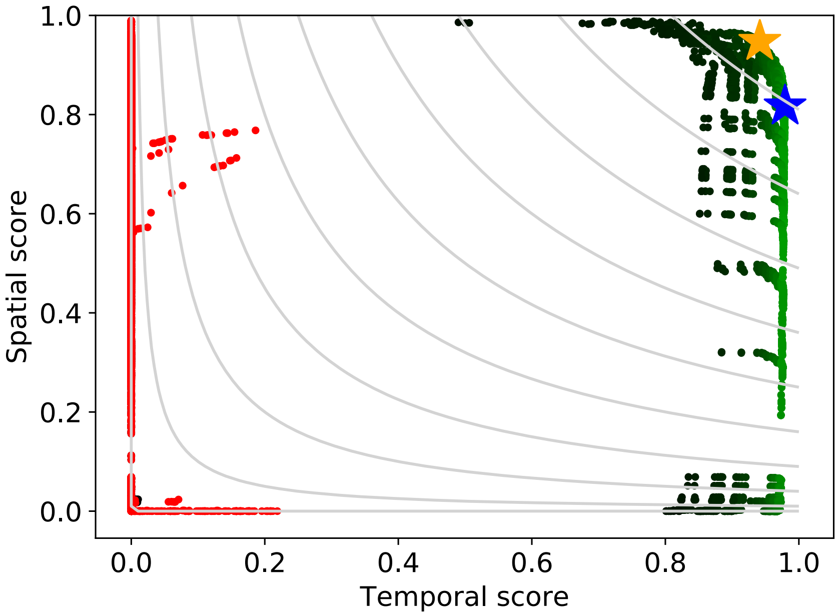
Analysis of the spatial and temporal score

The temporal evaluation score was plotted against the spatial evaluation score of all candidates as a scatter plot in Fig. 3.6. This work defined *possibly-correct relation* as a relation that both the prior and posterior event clusters are a subset of the ground-truth prior/posterior event clusters, excluding noise. The black and green dots indicated a possibly correct relation, shaded by the F-score of the relation from black (low F-score) to green (high F-score). The red dots indicated wrong relations. The scores of the orange mark, which is the relation with the highest evaluation score are shown in Table 3.4.

Fig. 3.6 indicated that the FDR-SC algorithm helped to clean up wrong relations and possibly-correct relations with low F-scores, while still keeping the possibly-correct relations with high F-scores intact. This results in a wider range of the temporal score



(a) G-CSM w/o FDR



(b) G-CSM w/ FDR-SC

Figure 3.6: Scatter plot of the temporal and spatial score. Red indicates the wrong relationship. Black to Green indicated possibly correct relations, shaded by F-score. The blue mark is the relation with the highest F-score, with orange being the highest evaluation score. Noise = 3000, $\lambda = 2$. Grey lines represented positions with equal final evaluation scores.

Table 3.4: Evaluation scores of the relation with the highest evaluation score from G-CSM

	Relation			Score	
Algo.	**Prec.**	**Rec.**	**F-Score**	**Temporal**[1]	**Spatial**
G-CSM w/o FDR	0.334	0.243	0.282	0.952	0.954
G-CSM w/ FDR-SC	0.331	0.270	0.298	0.942	0.947

[1] The temporal evaluation scores of G-CSM without FDR and with FDR-SC are calculated differently and cannot be compared directly.

over the pattern candidates. As shown in Fig. 3.6, the FDR-SC version has a usable range from around 0.6 to 1.0, while the without FDR is around 0.7 to 1.0. This can also be seen in Table 3.4, as the relation with the highest evaluation score also has a higher recall and F-score for the G-CSM with FDR-SC compared to G-CSM without FDR.

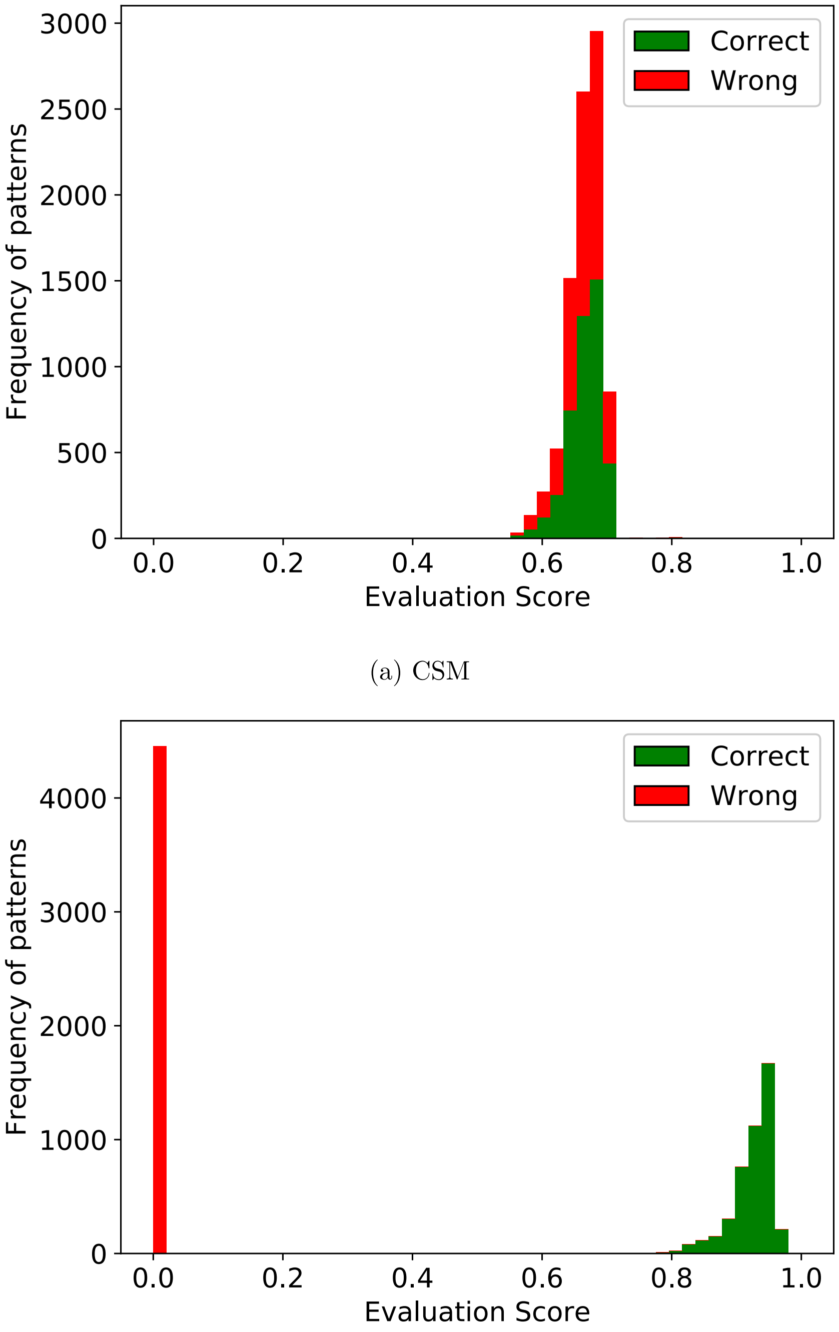
Therefore, the spatial score ended up having less effect on the final evaluation score. With the noisy data tested here — the higher the noise, the more noise was included in the candidate clusters — having less influence from the spatial evaluation allows the clusters to be bigger, thus a higher recall score. This can also be seen in the table, with FDR-SC having a higher recall score than the one without FDR.

3.5.4 Parameter analysis

Minimum sequence threshold $\mathcal{L}_{min}$

To analyze the effect of the minimum sequence threshold $\mathcal{L}_{min}$, the histogram of the final evaluation score of all valid cluster sequence patterns was plotted, and whether they are considered to be correct or wrong relations. The data used in this experiment was Noise = 100 and $\lambda = 2$. The other hyper-parameters were the same as in Section 3.5.3.

The resulting histogram is shown in Fig. 3.7, and it is clear that G-CSM has a very good separation between the evaluation score for correct and wrong relations, unlike the original CSM. It can be concluded that G-CSM is less sensitive to the $\mathcal{L}_{min}$ parameter. Note that $\mathcal{L}_{min} = 0.8$ and all the generated patterns by the CSM algorithm have scores less than 0.8, so CSM cannot output any patterns. According to Fig. 3.7, the detected relations have a final evaluation score of 0.7 or less.



(a) CSM

(b) G-CSM w/o FDR

Figure 3.7: Histogram of $\mathcal{L}$, CSM vs G-CSM. Noise = 100, $\lambda = 2$.

Table 3.5: CSM and G-CSM at various alpha settings. P = Precision, R = Recall, F = F-score, (1) = Prior cluster, (2) = Posterior cluster, (R) = Relation, Cnt. = Number of relations identified. Bold indicated the best result.

Algo.	**P (1)**	**R (1)**	**F (1)**	**P (2)**	**R (2)**	**F (2)**	**P (R)**	**R (R)**	**F (R)**	**Cnt.**
alpha = 0.001										
G-CSM	0.684	**0.632**	**0.650**	0.696	**0.627**	0.652	**0.428**	**0.399**	**0.410**	1.400
FDR-TH	0.657	0.499	0.537	0.679	0.555	0.594	0.329	0.279	0.295	2.950
FDR-SC	**0.695**	0.603	0.642	**0.704**	0.620	**0.654**	0.422	0.373	0.395	**1.300**
alpha = 0.005										
G-CSM	0.683	**0.633**	**0.650**	**0.700**	0.632	**0.659**	**0.431**	**0.401**	**0.413**	1.550
FDR-TH	0.663	0.492	0.539	0.676	0.552	0.592	0.330	0.277	0.295	3.000
FDR-SC	**0.685**	0.625	0.645	0.698	**0.632**	0.657	0.428	0.398	0.410	**1.450**
alpha = 0.01										
G-CSM	**0.685**	0.624	0.645	**0.700**	0.624	0.655	0.428	0.393	0.407	1.600
FDR-TH	0.606	0.509	0.528	0.612	0.545	0.563	0.304	0.274	0.283	3.250
FDR-SC	0.685	**0.627**	**0.647**	0.696	**0.641**	**0.662**	**0.431**	**0.404**	**0.414**	**1.500**
alpha = 0.05										
G-CSM	0.682	0.611	0.635	**0.703**	0.625	0.654	0.422	0.386	0.400	1.700
FDR-TH	0.452	0.519	0.457	0.462	0.560	0.481	0.234	0.284	0.245	4.050
FDR-SC	**0.685**	**0.625**	**0.646**	0.700	**0.630**	**0.658**	**0.428**	**0.394**	**0.408**	**1.600**
alpha = 0.1										
G-CSM	0.675	0.614	0.629	**0.702**	0.613	0.646	0.413	0.380	0.391	1.850
FDR-TH	0.375	0.612	0.426	0.400	**0.639**	0.455	0.222	0.392	0.264	4.950
FDR-SC	**0.682**	**0.619**	**0.640**	0.698	0.639	**0.662**	**0.428**	**0.398**	**0.410**	**1.700**
alpha = 0.2										
G-CSM	0.677	0.604	0.624	0.697	0.603	0.636	0.404	0.368	0.379	1.950
FDR-TH	0.251	**0.649**	0.348	0.296	**0.664**	0.389	0.169	**0.434**	0.235	7.050
FDR-SC	**0.682**	0.611	**0.635**	**0.701**	0.629	**0.657**	**0.423**	0.388	**0.401**	**1.700**

Significant threshold α

This work also investigated how changing the α significant threshold affected the result. Here, the same data as in the first experiment (Noise = 3000, $\lambda = 2$) were used with different values of α. The result is shown in Table 3.5. The G-CSM uses the specified alpha value directly to calculate the threshold in eq. (3.18).

With different significant threshold settings from 0.001 (0.1%) to 0.2 (20%), G-CSM with FDR-SC can maintain both the cluster and relation F-score better than the other algorithms. The result shows that G-CSM with FDR-SC is less sensitive to the alpha setting, and also has the number of extracted relations (Cnt.) close to one, which is better in this case.

3.5.5 Other type of patterns

The experiment in the previous section uses a single pair of events cluster. This section is to show that the proposed algorithm also worked with other types of data as well.

Four different types of patterns were tested as shown in Fig. 3.8. First, two relations were generated using the method described in Section 3.5.1, with Noise = 1000 and λ = 2, shown in Fig. 3.8a. The second data also has two relations, but the prior event locations were shared between two relations, shown in Fig. 3.8c. Third, the posterior cluster of one relation shared the location with the prior cluster of the second relation, shown in Fig 3.8e. And fourth, the variance of the prior and the posterior cluster were varied, shown in Fig. 3.8g. The timestamps of events are shifted by a uniform random number between 0 and 10,000, different for each relation.

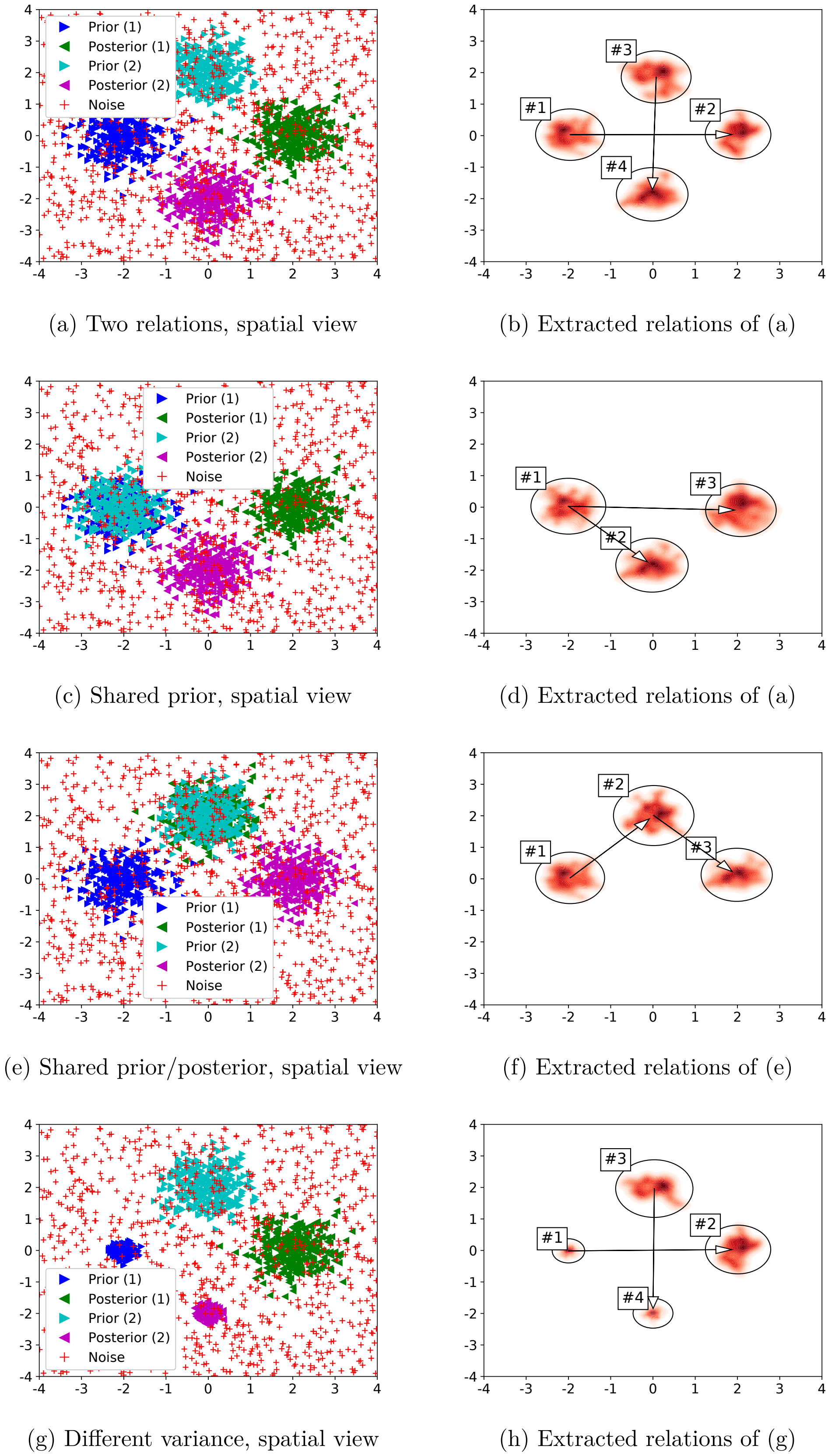
The hyper-parameters were the same as in Section 3.5.3. The extracted relations are shown in Fig. 3.8. The proposed G-CSM with the FDR-SC algorithm can correctly extract relations in all cases.

3.5.6 Semi-real data

The experiment in this section was using semi-real-world data. That is, real-world data is used as the spatial component, while the temporal component utilizes the same method as the synthetic generation.

The test data of UCI Machine Learning Optical Recognition of Handwritten Digits Data Set [44] was used as the spatial data. This represents a real-world example of spatial data that has predefined spatial clusters. The data contains 1,797 samples divided into 10 classes. Each class has approximately 180 samples. The data were preprocessed to normalize the mean and variance of each dimension (z-mean normalization). The input data with 64 dimensions were reduced to 10 dimensions using Neighborhood Components Analysis [45], as shown in Fig. 3.9a. This work randomly selected 2 pairs of digits as the embedded relations, while the other 6 digits' data were used as noise with uniform distribution over the temporal component. The interval between each embedded event pair of both relations followed exponential distribution with $\lambda = 2$. The parameters were the same as the previous experiment in Section 3.5.3. An example of what the timeline looks like is shown in Fig. 3.9b.

The result is shown in Table 3.6. In this data set, as there was no extra random noise added, even the original CSM, which is quite weak to spatial noise, can extract some relations, but it still performed worse than the other algorithms. FDR-SC performed the best in all evaluation metrics. Since the spatial dimension is reasonably well-separated and no random noise was added, the precision score was very high as the algorithm can easily extract the proper cluster. The recall score is limited by the spatial evaluation



(a) Two relations, spatial view

(b) Extracted relations of (a)

(c) Shared prior, spatial view

(d) Extracted relations of (a)

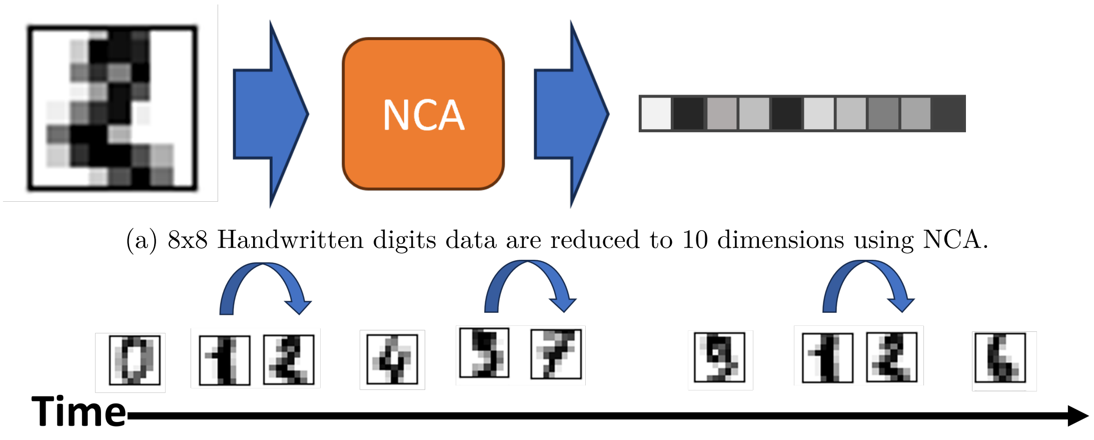
(e) Shared prior/posterior, spatial view

(f) Extracted relations of (e)

(g) Different variance, spatial view

(h) Extracted relations of (g)

Figure 3.8: Generated data and extracted relations using G-CSM with FDR-SC for other types of pattern



(a) 8x8 Handwritten digits data are reduced to 10 dimensions using NCA.

(b) Example of timeline with digits data

Figure 3.9: Semi-real experiments data detail

Table 3.6: CSM and G-CSM result from digits dataset. P = Precision, R = Recall, F = F-score, (1) = Prior cluster, (2) = Posterior cluster, (R) = Relation, Cnt. = Number of relations identified. Bold indicated the best result.

Algo.	P (1)	R (1)	F (1)	P (2)	R (2)	F (2)	P (R)	R (R)	F (R)	Cnt.
CSM	0.349	0.063	0.106	0.350	0.173	0.231	0.185	0.063	0.094	0.350
G-CSM	**0.997**	0.519	0.675	**1.000**	0.487	0.650	0.505	0.255	0.338	**2.000**
FDR-TH	**0.997**	0.495	0.658	0.989	0.457	0.622	0.476	0.228	0.308	2.050
FDR-SC	**0.997**	**0.532**	**0.687**	**1.000**	**0.497**	**0.658**	**0.517**	**0.267**	**0.351**	**2.000**

score, which prefers a compact cluster over a larger cluster. An adjustment to the hyper-parameters of the evaluation function may be needed to get a higher recall score, but otherwise, both the prior and posterior clusters of the extracted relations were the subset of those of the ground-truth relation. The scores on the relation evaluation are also low for a similar reason.

3.5.7 Complexity Analysis of the G-CSM algorithm

The original CSM algorithm has a run-time complexity of $\mathcal{O}(N^2 \log N)$ in the average case, where N is the number of data points. Within the algorithm, the time proximity of temporal evaluation is $\mathcal{O}(|A| + |B|)$ where $|A|$ and $|B|$ is the number of events in the prior and posterior cluster of each pattern, respectively.

For the proposed G-CSM algorithm, the time proximity algorithm uses GLM model fitting to calculate Granger causality strength. GLM model-fitting has runtime complexity of $\mathcal{O}(p^3 + Rp^2)$ where p is the number of predictors and R is the number of samples. In this case, $p = 2 \times M_i + 1$, which is a constant, and R is at most

$(2M_i + 1) \times (|A| + |B|) \sim (|A| + |B|)$, thus the GLM model fitting takes $\mathcal{O}(|A| + |B|)$, which is the same as original CSM. The FDR procedure takes $\mathcal{O}(N^2 \log N)$ in the worst case. Thus, G-CSM also has the runtime complexity of $\mathcal{O}(N^2 \log N)$, which is also the same as the original CSM.

3.5.8 Limitation of the G-CSM algorithm

Algorithm limitation

The G-CSM algorithm inherited all the limitations of the original Granger causality. The major point is that though Granger causality is one of the accepted methods to detect causality, it cannot be guaranteed whether it is an actual causality or not, just that it is causality under Granger's definition.

Since Granger measured the causality based on predictability, it is also limited by the predictor. In the case of G-CSM, the limitations of GLM models used as a predictor are the same as the traditional multivariate vector autoregressive (MVAR) model, mainly: linearity, stationarity, and dependency on observed variables. Moreover, since the proposed algorithm uses pairwise causality, more data from the environment might be missed, such as when two events are the cause of another event.

Complexity limitation

The algorithm complexity is $\mathcal{O}(N^2 \log N)$, a quadratic complexity, which can be a limiting factor with larger data sets. For example, the data used in the experiment section was limited to under 5,000 data points. Note that this was not the hard limit, but was a self-imposed limit to keep the running time reasonable. The author has experimented with 10,000 data points once (the earthquake data) but the algorithm was not able to complete it in a reasonable time frame.

In addition, there is a need to adjust the hyper-parameters of the algorithm to match the data used. With the long running time, adjusting the hyper-parameters is extremely time-consuming. This is the core limitation that can prevent application with actual real-world data.

3.6 Summary

This chapter proposed a Granger Cluster Sequence Mining (G-CSM) algorithm which is an extension of the original Cluster Sequence Mining (CSM) algorithm. It works by trying to find two spatial clusters in a point-process spatio-temporal data and try to detect if there was a causal relationship between the occurrence of events in both clusters using the Granger Causality procedure.

The experiments showed that the proposed algorithm has better relation extraction accuracy than the original CSM algorithm while keeping the same runtime complexity. Nevertheless, the complexity still hinders the application of real-world data.

Chapter 4

Local Density Estimation for Point Process Vector Autoregressive Model

4.1 Overview

The causality inference from the G-CSM algorithm uses a vector autoregressive (VAR) model with the GLM method to model the predictor. The process, while simple and fast to run, has drawbacks that a long history length cannot be easily used. To increase the history length of the predictor, either the number of windows must be increased, leading to more model parameters and longer running time, or the size of each time window must be expanded, which results in lower temporal resolution. Both are not ideal.

This chapter proposed a new procedure called *local density estimation*, which is a pre-processing step to modeling the VAR model. Specifically, instead of modeling the history of temporal point process data just by the presence of data (as in Fig. 3.3), this procedure instead performs a kernel density estimation over a fixed size of the temporal history and then applies auto-regression on the estimated density. This model is shown in Fig. 4.1. The procedure allowed the VAR model to better capture the precise location of each data in the point process, especially on sparse data, as well as allow easy scaling

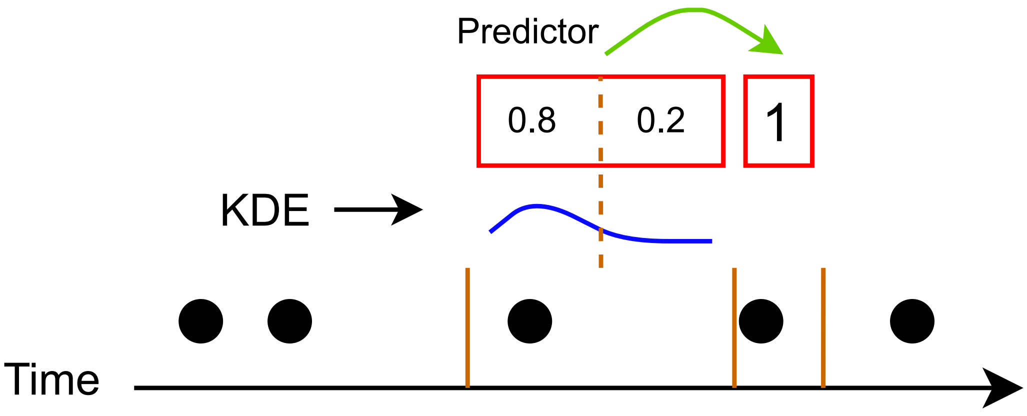


Figure 4.1: A model using local density estimation

to longer temporal history length by having a few parameters covering a long time span, while keeping the number of inputs to the model at a manageable level.

Using a linear and Gaussian kernel density model, this chapter described experiments with synthetic data generated with the Poisson model, which showed that the kernel-density pre-processing step improved the accuracy of prediction while still maintaining the same number of inputs.

4.2 VAR modeling of point process

Vector auto-regressive (VAR) is a model where a variable at the current time step is predicted by the past value of itself. For a general VAR model, consider a time-series $\mathbf{A} = \{a_0, a_1, \ldots, a_n\}, a_i \in \mathbb{R}$, the value of a_i can be modeled by:

$$a_i = \beta_0 + \sum_{j=1}^{k} \beta_j a_{i-j} + \varepsilon_i, \tag{4.1}$$

where k is the number of lagged variables, β are the model parameter, and ε is the error term.

A cumulative incidence function (CIF) is a core process of modeling a point process. The function indicates the rate of event occurrence at the specific time t parameterized by the history of event occurrence:

$$\lambda(t|H(t)) = \lim_{\Delta \to 0} \frac{Pr[(N(t+\Delta) - N(t)) = 1]}{\Delta}, \tag{4.2}$$

where $N(t)$ is a counting measure of the event within the time of $(0, t]$, and $H(t)$ is an occurrence history of all event occurrences up to time t. The probability of the event occurring in a small time window $[t, t + \Delta)$ can be written as $\lambda(t|H(t))\Delta$.

To use VAR to model a point process CIF, the timeline was divided into small slices of time windows. Then, take the number of events that occur in each slice of the window to be the value at each time step of the VAR model. More formally, consider a point process $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$ where x_i is a timestamp of each event in the point process. Let $T_0 = x_1$ and $T_1 = x_n$ be the minimum and maximum timestamp of the event, the whole timeline was divided into $K = (T_1 - T_0)/W$ slices of the window where W is the window size. Let R_i denote the number of occurrence of events in the time window $[T_0 + iW, T_0 + (i + 1)W)$, and $R(t)$ denote the R_i that is correspondent to the time t.

To model the incidence function, a generalized linear model (GLM) framework was used to model the CIF. In GLM, the logarithm of the CIF was modeled using a linear combination of the occurrence history:

$$\log \lambda(t|\theta, H(t)) = \theta_0 + \sum_{m=1}^{k} \theta_m R(t - mW), \tag{4.3}$$

where θ_0 is a background activity, and θ_m is the effect of $R(t)$.

A point process likelihood function [29] was used to fit the GLM model. This process uses a Generalized Linear Model (GLM) to fit the aforementioned log-CIF model. This is done by assuming Poisson distribution for $\lambda(t|\theta, H(t))$, because this target variable of the predictor can only be zero or positive integer. The GLM with Poisson distribution has the form of:

$$\log(\mu) = \eta = \beta_0 + \sum \beta_i x_i, \tag{4.4}$$

which is the same from as eq (4.3).

Note that the target variable can also be assumed to be in a binomial distribution (value is in $\{0, 1\}$), which would have a logistic function as the link function. However, [29] has proved that in case the target window only has 0 or 1 events, then both models are equivalent.

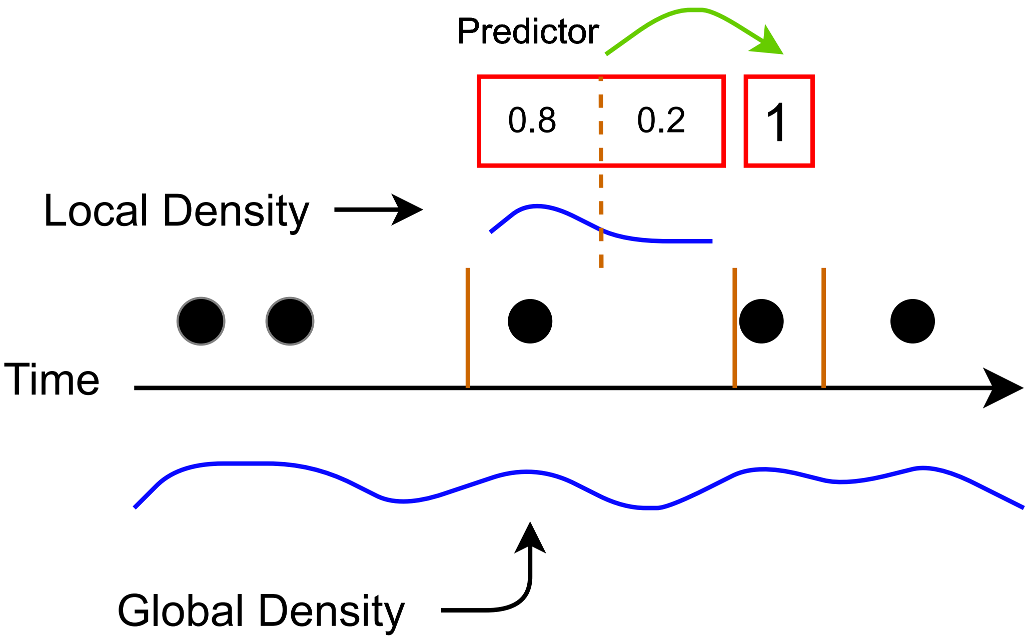


Figure 4.2: Local Density vs Global Density

4.3 Local density estimation

For the standard VAR model to capture longer history, there is a need to either 1) increase the number of history slices, or 2) increase the window size W. Both are not ideal: increasing the number of history slices results in an increased number of model parameters, which affect the runtime performance of the process; while increasing the size of the windows W results in reduced temporal accuracy. This can be problematic, especially in sparse data where a longer history length may be required.

To fix the aforementioned problems, this chapter introduced 1-dimensional kernel density pre-processing to the VAR model. Instead of using the lagged variable directly, the procedure sampled from a kernel density estimation trained on the event occurrence history of each prediction. Note that only event occurrence history relevant to each prediction was used for estimation to save on computational cost and to avoid information leakage from the predictor target. This allowed for increasing the history length of the model while keeping the number of model parameters low and still keeping some accuracy. The proposed procedure is hence named *local density estimation,* which is in contrast to global density where the density of every point was used. This difference is shown in Fig. 4.2.

Formally, given kernel K, bandwidth b, history length h, and the number of pa-

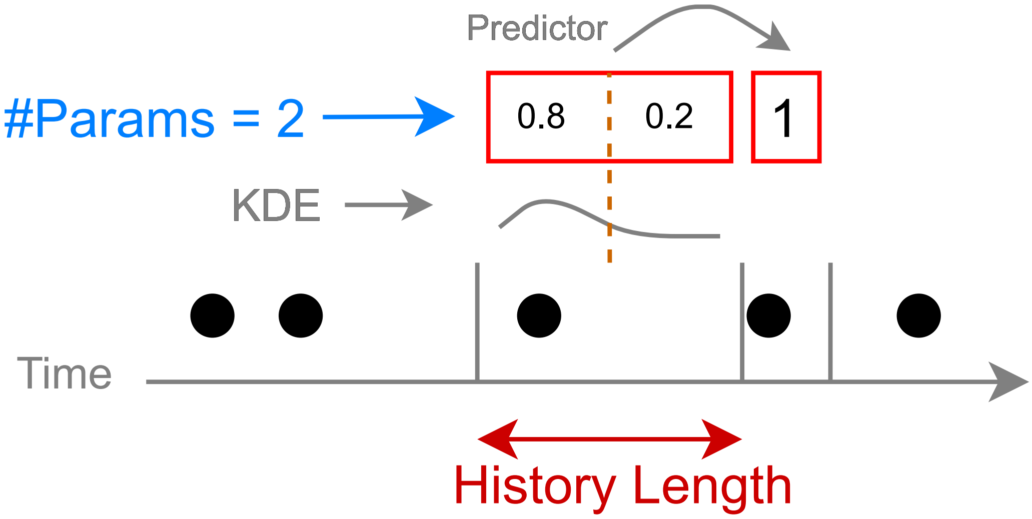


Figure 4.3: Difference between history length and number of parameters.

rameters p, to model CIF at the time window $[t, t+\Delta)$, the procedure first created a list of events during the time $[t-h, t)$, $\hat{\mathbf{X}} = \{x_i; t-h \leq x_i < t\}$. Then, the estimated density can be discretized to $\mathbf{D} = \{d_1, \ldots, d_k\}$ from the event list $\hat{\mathbf{X}}$ using kernel density estimation. The density at d_i can be calculated using the following formula:

$$d_i = \frac{1}{\hat{n}} \sum_{j=1}^{\hat{n}} K((t - \frac{ih}{k-1}) - \hat{x}_j, b), \tag{4.5}$$

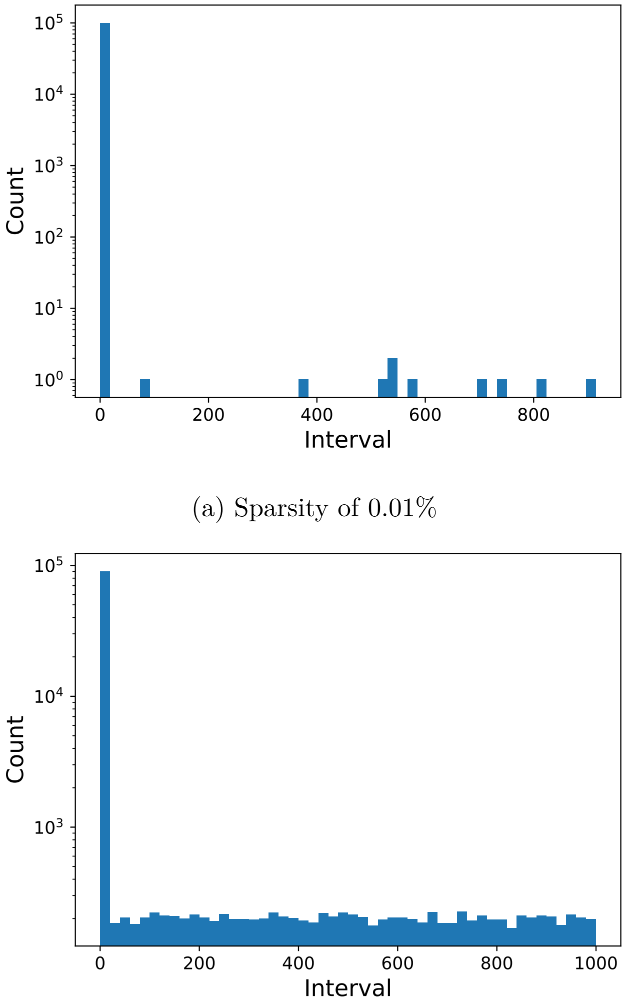
where $\hat{n} = \|\hat{\mathbf{X}}\|$. The discretized $\mathbf{D}$ is used instead of $R(t-mW)$ in eq. (3.10) for modeling a point process, yielding this new model. Note that the number of d_i may be lower than $R(t-mW)$.

$$\log \lambda(t|\theta, H(t)) = \theta_0 + \sum_{i=1}^{p} \theta_i d_i. \tag{4.6}$$

In this work, two types of kernel K were used: a linear (LIN) and a Gaussian (GAU) kernel:

$$K_{LIN}(x, b) \propto 1 - |x|/n \text{ if } |x| < b, \tag{4.7}$$

$$K_{GAU}(x, b) \propto \exp\left(-\frac{x^2}{2b^2}\right). \tag{4.8}$$



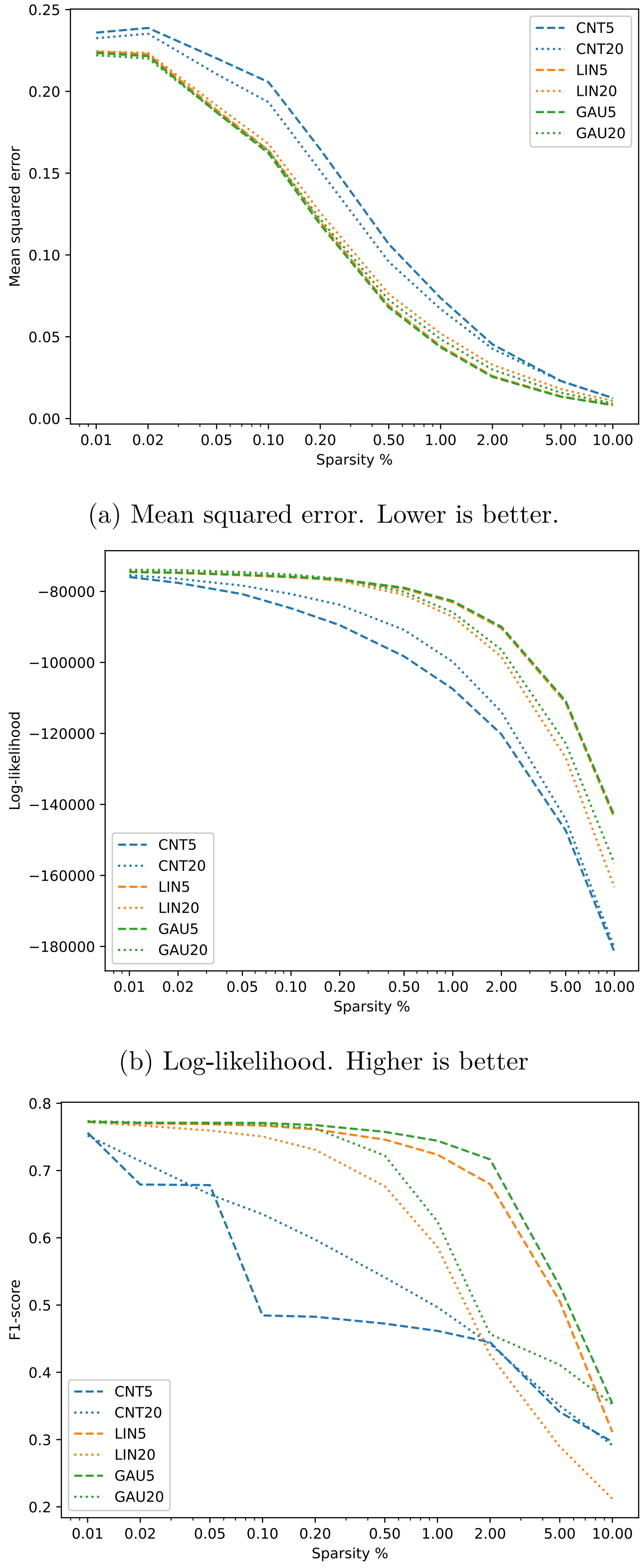
(a) Sparsity of 0.01%

(b) Sparsity of 10%

Figure 4.4: Histograms of the interval between events at different sparsity.

Table 4.1: Models used in the experiment

Name	**Kernel**	**History**	**#Params**	**Bandwidth**
CNT5	None	5	5	-
CNT20	None	20	20	-
LIN5	Linear	5	5	2
LIN20	Linear	20	5	5
GAU5	Gaussian	5	5	2
GAU20	Gaussian	20	5	5



(a) Mean squared error. Lower is better.

(b) Log-likelihood. Higher is better

(c) F1-score. Higher is better

Figure 4.5: Mean-squared error (MSE), log-likelihood, and F1-score of each model. CNT is a regular VAR model, while LIN and GAU are proposed methods with the linear and Gaussian kernel, respectively.

Table 4.2: Mean-squared error (MSE), log-likelihood, and F1-score of each model.

Sparsity	CNT5	CNT20	LIN5	LIN20	GAU5	GAU20
Mean squared error (lower is better)						
0.01%	0.2359	0.2325	0.2243	0.2245	0.2235	**0.2221**
0.02%	0.2388	0.2353	0.2227	0.2235	0.2215	**0.2201**
0.05%	0.2202	0.2106	0.1886	0.1913	**0.1872**	0.1874
0.1%	0.2057	0.1934	0.1640	0.1680	**0.1622**	0.1637
0.2%	0.1646	0.1514	0.1202	0.1259	**0.1186**	0.1218
0.5%	0.1066	0.0958	0.0690	0.0762	**0.0679**	0.0725
1%	0.0737	0.0670	0.0444	0.0520	**0.0435**	0.0485
2%	0.0454	0.0427	0.0260	0.0328	**0.0255**	0.0299
5%	0.0230	0.0228	0.0136	0.0181	**0.0133**	0.0158
10%	0.0125	0.0127	0.0083	0.0108	**0.0082**	0.0092
Log-likelihood (higher is better)						
0.01%	-75945	-75423	-74559	-74152	-74507	**-73906**
0.02%	-77597	-76431	-74807	-74295	-74723	**-73976**
0.05%	-80767	-78371	-75446	-74971	-75312	**-74574**
0.1%	-84755	-80725	-76038	-75738	-75852	**-75253**
0.2%	-89495	-83782	-76830	-77019	-76588	**-76438**
0.5%	-98297	-90825	-79260	-80970	**-78963**	-80132
1%	-107475	-99820	-83046	-87119	**-82695**	-85878
2%	-120250	-113927	-90357	-98409	**-89948**	-96410
5%	-147333	-144173	-111383	-126838	**-110882**	-122814
10%	-181505	-180393	-143822	-163314	**-143188**	-156588
F1-score (higher is better)						
0.01%	0.7562	0.7520	0.7722	0.7719	0.7726	**0.7735**
0.02%	0.6791	0.7141	0.7700	0.7670	0.7709	**0.7717**
0.05%	0.6784	0.6644	0.7690	0.7595	**0.7711**	0.7697
0.1%	0.4846	0.6351	0.7668	0.7506	**0.7708**	0.7690
0.2%	0.4826	0.5972	0.7612	0.7309	**0.7676**	0.7626
0.5%	0.4724	0.5411	0.7459	0.6767	**0.7575**	0.7222
1%	0.4616	0.4968	0.7236	0.5863	**0.7443**	0.6241
2%	0.4446	0.4420	0.6797	0.4259	**0.7166**	0.4567
5%	0.3410	0.3503	0.5057	0.2896	**0.5281**	0.4111
10%	0.2967	0.2916	0.3111	0.2120	**0.3537**	0.3526

4.4 Experiments

The proposed kernel-density pre-processing using linear and Gaussian kernel was tested against a regular vector autoregressive model using synthetic sparse Poisson process data. Mean-squared error of the prediction result, log-likelihood of the GLM model in eq. (3.10), and F1 score of each model were measured for comparison.

The synthetic data are regular Poisson process data that have the interval between each event occurrence followed by an exponential distribution:

$$\mathbf{L} = \{l_i \sim \text{Exp}(\lambda)\}, \tag{4.9}$$

$$\mathbf{X} = \{x_i = \sum_{j=0}^{i} l_j\}, \tag{4.10}$$

where λ is the exponential distribution mean. Sparsity was artificially added to this point process by randomly replacing s number of l_i with $g_i \sim \text{Uniform}[10, 1000]$. This created a random large gap within the timeline of the point process. This parameter s is called a sparsity count. This chapter uses $\lambda = 1$ for the Poisson process, which has an average interval of 1 and 90% the intervals are less than 3. Sparsity count s of 10 (0.01% of all data), 20 (0.02%), 50 (0.05%), 100 (0.1%), 200 (0.2%), 500 (0.5%), 1000 (1%), 2000 (2%), 5000 (5%), and 10000 (10% of all data) were used. The histograms of the interval between events are shown in Fig. 4.4.

100,000 points were generated for each point process, with 80,000 points being used for training and another 20,000 points for evaluation. The regular VAR model (denoted as CNT) was tested against the proposed model with a linear (triangle) kernel and Gaussian kernel (denoted as LIN and GAU). The detail was described in Table 4.1. The history length was the overall length of the history being used in each prediction, and the number of parameters described the number of inputs to the model. This is also shown in Fig. 4.3. All VAR models have a window size of 1. All models have a target window size of 1.

Each experiment was performed 10 times and the average of the result was taken. The MSE, the log-likelihood of the predictor, and the F1-score, calculated by thresholding the predictor output at 0.5, are shown in Fig. 4.5. In almost every case, the GAU5 model performed the best, followed closely by LIN5. The VAR model CNT5 and CNT20 perform worse in almost every case. Fig. 4.5c also shows that LIN5 and GAU5 are also less affected by the sparsity. GAU5 outperforms CNT20 with significantly fewer model parameters; GAU5 has only 5 parameters, whereas CNT20 utilizes 20. Note that as sparsity increases, the data can get extremely imbalanced so the MSEs are lower with higher sparsity.

4.5 Complexity analysis

Preparing a temporal point process data, especially for sparse data, for VAR modeling has the complexity of $\mathcal{O}(LN + N\log N)$ where L is the number of windows, W is the window size, and N is the number of data. This came from the following steps:

1. For each data point N:
    (a) Find the points that are in the history length ($\mathcal{O}(\log N)$ using binary search)
    (b) Construct a history model from at most N points ($\mathcal{O}(N)$)
2. However, in Step 1b, note that all points can be part of at most L history models. Hence, step 1b amortized to $\mathcal{O}(LN)$

Step 1, minus the amortized part, has the complexity of $\mathcal{O}(N\log N)$. The amortized part is $\mathcal{O}(LN)$, yielding the final complexity of $\mathcal{O}(LN + N\log N)$.

For the proposed local density estimation, the complexity is $\mathcal{O}(\frac{h}{w}N + Np + N\log N)$ where h is the history length, p is the number of history samples (number of parameters), and w is the time step used for the prediction target. Similarly, this came from:

1. For each data point N:
    (a) Find the points that are in the history length ($\mathcal{O}(\log N)$ using binary search)
    (b) Construct and sample p samples of density from at most N points ($\mathcal{O}(N+p)$)
2. However, in Step 1b, again, all points can be part of at most $\frac{h}{w}$ history models. Hence, step 1b amortized to $\mathcal{O}(\frac{h}{w}N + Np)$

Note that $\frac{h}{w}$ is essentially L in the complexity of the regular VAR model. Hence, the proposed algorithm can only be asymptotically slower than the regular VAR model if and only if Np is larger than both $\frac{h}{w}N$ and $N\log N$, which seems unlikely, as part of the reason to use this procedure is to reduce the number of model parameter p to be less than $\frac{h}{w}$.

4.6 Summary

This chapter detailed the *local density estimation* procedure for increasing the history length of the vector autoregressive model while keeping the model parameter low. This worked by applying a 1-dimensional kernel density estimation over the event history to be used for prediction.

The experiments showed that the works well under sparse data, and can beat regular models even with fewer parameters. However, there are limitations to this method. Mainly, the vector autoregressive method may not be a good model for specific data in the first place.

Chapter 5

Conclusion

5.1 Summary

Spatio-temporal data analysis is already harder than just spatial data analysis or temporal data analysis. Though many techniques have been developed, the field is still very young, despite a large amount of real-world spatio-temporal data.

While there were some developed methods for spatio-temporal relationship mining (frequent pattern mining) and spatio-temporal change detection, there were none that are designed specifically for *detecting a change in spatio-temporal relation*. Spatio-temporal relations affect a lot of natural and man-made phenomenon data around us, including weather systems or crime analysis.

This work first proposed changes to the existing algorithm for spatio-temporal occurrence correlation detection technique called Cluster Sequence Mining (CSM) with the added Granger causality measurement. The result, the Granger Cluster Sequence Mining (G-CSM) algorithm, is an algorithm for the detection of causal relations in spatio-temporal data. The experiment shows that G-CSM can better detect and is much more resilient than the original CSM technique. False Discovery Rate (FDR) further improved the result.

The local density estimation procedure was also proposed. This procedure allows increasing the history length of the predictor used during causality inference while keeping the runtime performance. The result showed that this procedure improved the predictor performance significantly, even outperforming the standard VAR model that has a higher number of model parameters. While this procedure is designed specifically for the G-CSM algorithm, it can be used in any instance where the VAR model is used with the point-process data.

5.2 Future Works

5.2.1 Applying Local Density Estimation to the G-CSM algorithm

The proposed Local Density Estimation algorithm in Chapter 4 is directly designed to be used as a part of the G-CSM algorithm to allow the model to capture longer history length while still keeping the runtime performance. This is the immediate future work, and implementation and experiments are required.

5.2.2 Spatial Clustering Improvement on G-CSM algorithm

The G-CSM algorithm is limited by the performance of the CSM algorithm it is based on. One of the desired improvements is the improvement of the spatial clustering process. Currently, the AHC algorithm in the CSM works well enough, but on dense spatial data, it can fail to capture the proper spatial clusters that are part of the causality. Since the AHC algorithm is hierarchical, there is a possibility that a ground-truth spatial cluster may not be a part of clusters that AHC found at all.

The challenge in this part is that the number of spatial clusters directly correlates to the number of candidates to be evaluated. The number of candidates is quadratic of the number of spatial clusters, hence, replacing this directly with other types of clustering algorithm is not feasible. A complete rethinking of the entire CSM algorithm may be required.

5.2.3 Extension to Non-Stationery Relations

In this work, it is assumed that the relations are stationary. That is, the spatial location of each cluster is constant in time. This may not be the case in real-world situations, where the spatial location can drift over time. The proposed algorithm currently cannot handle the non-stationary relation at all.

5.2.4 Non-Linear Extension fo Causality Detection

One of the limitations of the current approach to the Granger causality inference method is that even with Local Density Estimation added, the predictor model is still linear. As previously discussed, there was a lot of ongoing research on modeling a point process data using various techniques, but the core problem is that: to check for causality, a likelihood must be calculated. A good deal of models, especially those involving neural networks, do not have a closed-form solution for the likelihood value of the model, requiring numerical integration to calculate the value.

In addition, runtime complexity must also be considered. Unlike most applications of point process or causality inference, the G-CSM algorithm performs causality inferences thousands if not millions of times. Currently, the VAR model with GLM is very fast, hence the algorithm can run in a reasonable time. Any more complex model can and will extend the runtime of the G-CSM by multiple magnitudes. As many candidates are superset/subset of each other, a possible solution might be a causality inference technique that can handle incremental processing or can handle multiple candidates at the same time.

5.2.5 Performance Improvement to the G-CSM algorithm

The main performance problem of the G-CSM algorithm is currently the number of candidates generated. This is, in general, quadratic to the number of spatial clusters, as discussed in Sec. 5.2.2. An improvement to the candidate generation process or the spatial cluster generation process, for example, by quickly eliminating candidates, would be very beneficial to the performance. Possible solutions might include a prior quick testing of causality, or, to completely rethink the process, an incremental causality inference technique that works directly without requiring candidate generation.

5.2.6 Other time interval distribution

This work currently mainly deals with the Poisson distribution, however, extension to other types of interval distribution should be possible. Note that as Poisson distribution usually results in the shortest interval between the cause and effect, the currently short history length of the predictor can capture the relationships. To model other distributions that may be longer, a longer history length is needed. With the local density estimation procedure, it should be possible but has not been tested yet.

5.2.7 Application to real-world usage

It was hoped that this algorithm would work on earthquake data to try to gain an understanding of the relations between each earthquake location. However, due to run time limitations, the need to adjust hyper-parameters, and the size of the earthquake dataset, this work did not complete such experiments. The number of earthquakes occurring around the Japanese archipelago is more than 10,000 occurrences per year, and while the number can be decreased by only looking at stronger earthquakes, one questioned the benefit of analyzing only such information.

However, in the long term, it is hoped that this algorithm can be used to find causal relations between different concepts in various settings including cross-domain applications like, for example, relationships between disease outbreaks and financial markets. This could enable humanity to get a better understanding of the various mechanics of nature or society.

Bibliography

[1] G. Atluri, A. Karpatne, and V. Kumar, “Spatio-temporal data mining: A survey of problems and methods,” *ACM Computing Surveys*, vol. 51, no. 4, pp. 1–41, 2018.

[2] T. M. Thompson, S. Rausch, R. K. Saari, and N. E. Selin, “A systems approach to evaluating the air quality co-benefits of us carbon policies,” *Nature Climate Change*, vol. 4, p. 917–923, Oct. 2014.

[3] L. Tompson, S. Johnson, M. Ashby, C. Perkins, and P. Edwards, “Uk open source crime data: accuracy and possibilities for research,” *Cartography and Geographic Information Science*, vol. 42, no. 2, pp. 97–111, 2015.

[4] P. S. Castro, D. Zhang, C. Chen, S. Li, and G. Pan, “From taxi gps traces to social and community dynamics: A survey,” *ACM Computing Surveys*, vol. 46, pp. 17:1–17:34, Dec. 2013.

[5] Y. Matsubara, Y. Sakurai, W. G. Van Panhuis, and C. Faloutsos, “Funnel: automatic mining of spatially coevolving epidemics,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York New York USA), p. 105–114, ACM, Aug. 2014.

[6] A.-K. Mahlein, “Plant disease detection by imaging sensors – parallels and specific demands for precision agriculture and plant phenotyping,” *Plant Disease*, vol. 100, p. 241–251, Feb. 2016.

[7] S. Shekhar, Z. Jiang, R. Ali, E. Eftelioglu, X. Tang, V. Gunturi, and X. Zhou, “Spatiotemporal data mining: A computational perspective,” *ISPRS International Journal of Geo-Information*, vol. 4, no. 4, pp. 2306–2338, 2015.

[8] J. A. González, F. J. Rodríguez-Cortés, O. Cronie, and J. Mateu, “Spatio-temporal point process statistics: A review,” *Spatial Statistics*, vol. 18, p. 505–544, Nov 2016.

[9] F. Musmeci and D. Vere-Jones, “A space-time clustering model for historical earthquakes,” *Annals of the Institute of Statistical Mathematics*, vol. 44, p. 1–11, Mar 1992.

[10] Z. Zhou, D. S. Matteson, D. B. Woodard, S. G. Henderson, and A. C. Micheas, “A spatio-temporal point process model for ambulance demand,” *Journal of the American Statistical Association*, vol. 110, p. 6–15, Jan 2015.

[11] A. Reinhart, “A review of self-exciting spatio-temporal point processes and their applications,” *Statistical Science*, vol. 33, no. 3, p. 299–318, 2018.

[12] R. T. Q. Chen, B. Amos, and M. Nickel, “Neural spatio-temporal point processes,” *International Conference on Learning Representations*, 2021.

[13] S. Li, S. Xiao, S. Zhu, N. Du, Y. Xie, and L. Song, “Learning temporal point processes via reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[14] K. Zhang and P. Spirtes, “Review of causal discovery methods based on graphical models,” *Frontiers in Genetics*, vol. 10, 06 2019.

[15] C. W. J. Granger, “Investigating causal relations by econometric models and cross-spectral methods,” *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.

[16] T. D. Le, T. Hoang, J. Li, L. Liu, H. Liu, and S. Hu, “A fast pc algorithm for high dimensional causal discovery with multi-core pcs,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 5, pp. 1483–1495, 2019.

[17] L. Jiang and L. Bai, “Spatio-temporal characteristics of urban air pollutions and their causal relationships: Evidence from beijing and its neighboring cities,” *Scientific Reports*, vol. 8, no. 11, p. 1279, 2018.

[18] I. Davidson, S. Gilpin, O. Carmichael, and P. Walker, “Network discovery via constrained tensor analysis of fmri data,” in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD ’13, pp. 194–202, ACM Press, 2013.

[19] K. Fukui, D. Inaba, and M. Numao, “Discovery of damage patterns in fuel cell and earthquake occurrence patterns by co-occurring cluster mining,” *Proc. The 2014 AAAI Workshop for Discovery Informatics*, pp. 19–26, 2014.

[20] K. Fukui, Y. Okada, K. Satoh, and M. Numao, “Cluster sequence mining from event sequence data and its application to damage correlation analysis,” *Knowledge-Based Systems*, vol. 179, pp. 136–144, 2019.

[21] *Spatial Point Processes and their Applications*, p. 1–75. Lecture Notes in Mathematics, Berlin, Heidelberg: Springer, 2007.

[22] D. R. Cox and V. Isham, *Point processes.* Monographs on applied probability and statistics, London; New York: Chapman and Hall, 1980.

[23] Y. Ogata, “Space-time point-process models for earthquake occurrences,” *Annals of the Institute of Statistical Mathematics*, vol. 50, p. 379–402, Jun 1998.

[24] Y. Ogata and J. Zhuang, “Space–time etas models and an improved extension,” *Tectonophysics*, vol. 413, p. 13–23, Feb 2006.

[25] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, “Recurrent marked temporal point processes: Embedding event history to vector,” *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 1555–1564, Aug 2016.

[26] M. Okawa, T. Iwata, T. Kurashima, Y. Tanaka, H. Toda, and N. Ueda, “Deep mixture point processes: Spatio-temporal event prediction with rich contextual information,” *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 373–383, Jul 2019.

[27] M. Higuchi, K. Matsutani, M. Kumano, and M. Kimura, “Discovering spatio-temporal latent influence in geographical attention dynamics,” *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2018)*, vol. 11052, p. 517–534, 2019.

[28] S. Zhu, S. Li, Z. Peng, and Y. Xie, “Interpretable deep generative spatio-temporal point processes,” 2020. AI for Earth Sciences Workshop at NeurIPS.

[29] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, “A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects,” *Journal of Neurophysiology*, vol. 93, no. 2, pp. 1074–1089, 2005.

[30] K. G. Pillai, R. A. Angryk, J. M. Banda, M. A. Schuh, and T. Wylie, "Spatio-temporal co-occurrence pattern mining in data sets with evolving regions," in *IEEE 12th International Conference on Data Mining Workshops*, pp. 805–812, 2012.

[31] Y. Huang, L. Zhang, and P. Zhang, "A framework for mining sequential patterns from spatio-temporal event data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, pp. 433–448, 2008. 148.

[32] Z. Zhou and D. S. Matteson, "Predicting ambulance demand: a spatio-temporal kernel approach," *Proc. The 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2297–2303, 2015.

[33] O. Sporns and R. Kötter, "Motifs in brain networks," *PLOS Biology*, vol. 2, no. 11, p. e369, 2004.

[34] P. Spirtes, C. Glymour, and R. Scheines, *Causation, Prediction, and Search, Second Edition.* A Bradford Book, Jan. 2001.

[35] J. Geweke, "Measurement of linear dependence and feedback between multiple time series," *Journal of the American Statistical Association*, vol. 77, no. 378, pp. 304–313, 1982.

[36] S. Kim, D. Putrino, S. Ghosh, and E. N. Brown, "A granger causality measure for point process models of ensemble neural spiking activity," *PLOS Computational Biology*, vol. 7, no. 3, p. e1001110, 2011.

[37] T. Schreiber, "Measuring information transfer," *Physical Review Letters*, vol. 85, pp. 461–464, 2000.

[38] L. Barnett, A. B. Barrett, and A. K. Seth, "Granger causality and transfer entropy are equivalent for gaussian variables," *Physical Review Letters*, vol. 103, p. 238701, 2009.

[39] B. Zarebavani, F. Jafarinejad, M. Hashemi, and S. Salehkaleybar, "cupc: Cuda-based parallel pc algorithm for causal structure learning on gpu," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 530–542, 2020.

[40] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen, "A linear non-gaussian acyclic model for causal discovery," *Journal of Machine Learning Research*, vol. 7, pp. 2003–2030, 2006.

[41] G. King, *Unifying Political Methodology: The Likelihood Theory of Statistical Inference*. Ann Arbor: University of Michigan Press, Jun 1998.

[42] Y. Benjamini and Y. Hochberg, “Controlling the false discovery rate: A practical and powerful approach to multiple testing,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 57, no. 1, p. 289–300, 1995.

[43] H. Akaike, “A new look at the statistical model identification,” *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[44] D. Dua and C. Graff, “UCI machine learning repository,” 2019.

[45] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, “Neighbourhood components analysis,” *Advances in Neural Information Processing Systems 17*, pp. 513–520, 2005.