STATISTICAL INFERENCE, MODELING, AND LEARNING OF POINT PROCESSES

A Dissertation
Presented to
The Academic Faculty

By

Shuang Li

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial & Systems Engineering

Georgia Institute of Technology

August 2019

STATISTICAL INFERENCE, MODELING, AND LEARNING OF POINT PROCESSES

Approved by:

Dr. Yao Xie, Advisor
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Le Song, Advisor
School of Computational Science & Engineering
Georgia Institute of Technology

Dr. Jianjun Shi
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Kamran Paynabar
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Dr. Yajun Mei
H. Milton Stewart School of Industrial & Systems Engineering
Georgia Institute of Technology

Date Approved: June 20, 2019

You can have data without information, but you cannot have information without data.

— Daniel Keys Moran

To my parents.

ACKNOWLEDGEMENTS

My foremost and deepest gratitude goes to my advisors, Yao Xie and Le Song. I have been very fortunate to work with them on exciting research topics. I am very grateful to their generous support and invaluable guidance, which led me to discover the beauty of research. It is hard to imagine this thesis without their guidance and support. I am constantly amazed by their passion for research, and insights and vision into problems. They can always find the perfect balance between theory and practice, from which I benefit a lot. I cannot overstate my appreciation for their constant availability and encouragement. I owe them a debt of gratitude larger than I can express here.

I am grateful to my friends and collaborators at and outside Georgia Tech for their friendship and for making my life wonderful. I appreciate all their accompany, suggestions and support. Special thanks go to Fang Cao, Junzhuo Chen, Minshuo Chen, Shanshan Cao, Yang Cao, Yilun Chen, Yufeng Cao, Zhehui Chen, Bo Dai, Hanjun Dai, Nan Du, Juan Du, Chen Feng, Mehrdad Farajtabar, Rui Gao, Junqi Hu, Haoming Jiang, Zhou Lan, Rakshit Trivedi, Rui Tuo, Wenjia Wang, Xing Wang, Guanyi Wang, Qianyi Wang, Bo Xie, Liyan Xie, Weijun Xie, Chuanping Yu, Xiaowei Yue, Can Zhang, Rui Zhang, Ruizhi Zhang, Shixiang Zhu, Wanrong Zhang, and Yi Zhou.

Last but not least, I want to thank my parents for bringing me to this world and for their unconditional love. To them, I dedicate this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Complex systems, such as healthcare systems, cities, and information networks, often produce a large volume of time series data, along with ordered event data, which are discrete in time and space, and rich in other features (e.g., markers or texts). We can model the asynchronous event data as point processes.

It is essential to understand and model the complex dynamics of these time series and event data so that accurate prediction, reliable detection, or smart intervention can be carried out for social goods. Specifically, my thesis focuses on the following aspects: (1) new statistical models and effective learning algorithms for complex dynamics exhibited in event data; (2) new inference algorithms for change-point detection, and temporal logic reasoning involving time series and event data.

In the first part of the thesis, we focus on the inference algorithms for change-point detection. We consider two settings to detect the changes. One is for high-dimensional streaming data, and the other is for networked asynchronous event data.

In the high-dimensional streaming data setting, we propose a kernel-based nonparametric change-point detection method, which enjoys fewer assumptions on the distributions. Theoretical tail probability approximation of the nonparametric statistic is also proposed, which provides a statistically principled way to determine the detection thresholds. The proposed nonparametric method shows excellent performance on real human-activity detection dataset and speech dataset.

In the networked asynchronous event data setting, we model the event data as point processes and propose a continuous-time change-point detection framework to detect dynamic changes in networks. Specifically, we cast the problem into a sequential hypothesis test, and derive the generalized likelihood-ratio (GLR) statistic for networked point processes by considering the network topology. The constructed statistic can achieve weak signal detection by aggregating local statistics over time and networks. We further propose to evaluate the proposed GLR statistic via an efficient EM-like algorithm which can be implemented in a distributed fashion across dimensions. Similarly, we obtain a highly accurate theoretical threshold characterization for the proposed GLR statistic and demonstrate the excellent performance of our method on real social media datasets, such as Twitter and Memetracker.

In the second part of the thesis, we focus on new statistical models and effective learning algorithms for point processes under the big data setting and the small data setting, respectively.

For the big data setting, we propose an highly expressive model for point processes and want the data to speak for themselves. Specifically, we leverage recent advances in deep learning and parameterize the intensity function of point processes as a recurrent neural network (RNN). RNN is a composition of a series of highly flexible nonlinear functions, which allows the model to capture complex dynamics in event data. Fitting neural network models for even data is challenging. We develop a novel adversarial learning framework to address this challenge and further avoid model-misspecification. The proposed framework has been evaluated on real crime, social network, and healthcare datasets, and outperforms the state-of-the-art methods in data description.

For the small data setting, we propose a unified framework to incorporate domain knowledge to point process models. The proposed temporal logic point processes model the intensity function of the event starts and ends via a set of first-order temporal logic rules. Using softened representation of temporal relations, and a weighted combination of logic rules, our framework can also deal with uncertainty in event data. We derive a maximum likelihood estimation procedure for the proposed temporal logic point processes, and show that it can lead to accurate predictions when data are sparse and domain knowledge is critical. The proposed framework has been evaluated on real healthcare datasets, and outperforms the neural network models in event predication on small data and is easy to interpret.

CHAPTER 1
INTRODUCTION

Social goods, such as healthcare, smart city and information networks, often produce a list of ordered event data with rich information in time, location and other features (e.g., text), i.e.,

$$\mathcal{H}_t := \{e_1 = (t_1, s_1, \kappa_1),\, e_2 = (t_2, s_2, \kappa_2),\, \dots,\, e_n = (t_n, s_n, \kappa_n)\},$$

where $t_i \in \mathbb{R}^+$ is the occurrence time of event $i \in \mathbb{Z}$, $s_i \in \mathbb{S}$ is the occurrence location and $\kappa_i \in \mathbb{M}$ is the associated feature.

It is essential to understand the complex dynamics of these event data and model the intricate spacetime-intertwined dynamics so that accurate prediction, detection or intervention can be carried out subsequently depending on the context. Use crime event as an example. Police departments worldwide are eager to develop better police resource allocation methods to manage the complex and evolving crime landscape. An accurate crime prediction model is the prerequisite for effective crime prevention, response and investigation.

However, for crime event and many other types of event data, the modeling and prediction face many challenges due to the irregular nature of the observation, the complex spatial, temporal and relational dynamics, and the additional high dimensional event markers or features. All these challenges together make the event data modeling a nontrivial problem.

Many existing approaches in dealing with event data usually require discretizing the time and space, and use some ad-hoc aggregations to convert the events to standard time-series sequences. This discretization and aggregation procedure, however, might not pool data efficiently or lose original information in events. Point processes offers an elegant mathematical framework for directly modeling the event data in continuous time and space. Classic temporal marked point process models the generative processes of events by conditional intensity function, defined as

$$\lambda(t, \kappa \,|\, \mathcal{H}_t) dt d\kappa = \mathbb{E}[N(dt \times d\kappa) \,|\, \mathcal{H}_t],$$

where $N(A)$ the number of (t_i, κ_i) falling in a set $A \subset \mathbb{R}^+ \times \mathbb{M}$.

The conditional intensity function specifies how the mean number of events in a region depends on the past in an evolutionary point process, and is hand-crafted by a parametric or nonparametric form to capture the potentially complex triggering and clustering pattern of events.

Suppose a parametric model $\lambda_\theta(t, \kappa \,|\, \mathcal{H}_t)$ has been specified by an unknown parameter θ, then using the maximum-likelihood-estimation (MLE) learning paradigm one can learn the model by maximizing the joint probability for a realization of $\{e_1, \ldots, e_n\}$ in terms of θ , i.e.,

$$\lambda_\theta^* = \arg\, \max\, L(\theta) := \exp\left\{ - \int \int_{(0,t) \times \mathbb{M}} \lambda_\theta(t, \kappa | \mathcal{H}_t) dt \right\} \prod_{i=1}^{n} \lambda_\theta(t_i, \kappa_i \,|\, \mathcal{H}_{t_i}).$$

The descriptive power of the estimated model relies heavily on expressiveness and flexibility of the intensity function.

Specifically, under the principled theoretical framework, my thesis focuses on the following aspects:

1. Novel inference algorithms for anomaly detection involving *time series* and *event* data (Chapter 2 and Chapter 3).

2. Novel statistical models and effective learning algorithms for complex dynamics exhibited in *event* data (Chapter 4 and Chapter 4);

CHAPTER 2

SCAN B-STATISTIC FOR KERNEL CHANGE-POINT DETECTION

Detecting the emergence of an abrupt change-point is a classic problem in statistics and machine learning. Kernel-based nonparametric statistics have been used for this task, which enjoys fewer assumptions on the distributions than the parametric approach and can handle high-dimensional data.

In this chapter, we focus on the scenario when the amount of background data is large, and propose a computationally efficient kernel-based statistics for change-point detection, which are inspired by the recently developed B-statistics. A novel theoretical result of the paper is the characterization of the tail probability of these statistics using the change-of-measure technique, which focuses on characterizing the tail of the detection statistics rather than obtaining its asymptotic distribution under the null distribution.

Such approximations are crucial to controlling the false alarm rate, which corresponds to the average-run-length in online change-point detection. Our approximations are shown to be highly accurate. Thus, they provide a convenient way to find detection thresholds for online cases without the need to resort to the more expensive simulations. We show that our methods perform well on both synthetic data and real data.

2.1 Overview

Given a sequence of samples, $x_1, x_2, \ldots, x_t$, from a domain $\mathcal{X}$, we are interested in detecting a possible change-point τ, such that before the change samples x_i are *i.i.d.* with a null distribution P, and after the change samples x_i are *i.i.d.* with a distribution Q. Here, we consider two scenarios: the time horizon t is fixed, $t = T_0$, which we call the offline or fixed-sample change-point detection, or the time horizon t is not fixed, meaning that one can keep getting new samples, which we call the online or sequential change-point detection. In the offline setting, our goal is to detect the existence of a change. In the online setting, our goal is to detect the emergence of a change as soon as possible after it occurs. Here, we restrict our attention to detecting one change-point. One such instance is seismic event detection as studied by [1], where one would like to either detect the presence of a weak event in retrospect to better understand the geophysical structure or detect the event as quickly as possible for online monitoring.

Ideally, the detection algorithm should be free of distributional assumptions to be robust when applied to real data. To achieve this goal, various kernel-based nonparametric statistics have been proposed in the statistics and machine learning literature, see, e.g., [2, 3, 4, 5, 6, 7], which typically work well with multi-dimensional real data since they are distributional free. Kernel approaches are distribution free and more robust as they provide consistent results over larger classes of data distributions; albeit they can be less powerful in settings where a clear distributional assumption can be made. However, most kernel based statistics cost $\mathcal{O}(n^2)$ to compute over n samples. In the online change-point detection setting, the number of samples grows with time and hence we cannot directly use the naive approach. Recently, [8] developed the so-called *B-test statistic* to reduce the computational complexity. The B-test statistic samples N pairs of blocks of size B from the two-sample data, compute the unbiased estimates of the kernel-based statistic between each pair and then take an average. The computational complexity of the B-test statistic reduces to $\mathcal{O}(nB^2)$ instead of $\mathcal{O}(n^2)$.

In this chapter, we present two scan statistics related to B-test statistics customized for offline and online change-point detection, which we name as *scan B-statistics*. The proposed statistics are based on kernel maximum mean discrepancy (MMD) in [9, 10]. They are inspired by the B-test statistic but differ in various ways to tailor to the need of change-point detection. Typically, there is a small number of post-change samples (for instance, seismic events are relatively rare, and in online change-point detection, one would like to detect the change quickly). But there is a large amount of reference data. So when constructing the detection statistic, we reuse the post-change samples for the test block and construct multiple and disjoint reference blocks. This leads to a non-negligible dependence between the MMD statistics being averaged over. Hence, we cannot use the existing approach based on the central limit theorem to analyze them. Moreover, the scanning nature of the proposed statistic also introduces non-negligible dependence. We construct the reference and test blocks in a structured way so that analytical expressions for false alarm can be obtained.

Our main theoretical contribution includes accurate theoretical approximations to the false-alarm rate of scan B-statistics. Controlling false alarms is a key challenge in change-point detection. Specifically, this means to quantify the significance level for offline change-point detection, and the average run length (ARL) for online change-point detection. Here, we cannot directly rely on the null property of the B-test statistic established in the existing work, because the scan statistics take the maximum of multiple statistics computed over overlapping data blocks that causes strong correlations. Hence, one cannot use the central limit theorem or even the martingale central limit theorem. Instead, we adopt a recently developed change-of-measure technique by [11] for scan statistics, which are capable of dealing with the more challenging situation here.

Our contribution also includes: (1) obtaining a closed-form variance estimator, which allows easy calculation of the scan B-statistics; (2) further improving the accuracy of our approximations by taking into account the skewness of the kernel-based statistics. The accuracy of our approximations is validated by numerical examples. Finally, we demonstrate the good performance of our method using real-data, including speech and human activity data.

2.1.1 Related work

Classic parametric approaches for change-point detection can be found in [12, 13]. There has been an array of nonparametric change-point detection methods. Notable non-parametric schemes for change-point detection include [14, 15], which are designed for scalar observations and not suitable for vector observations. [16] provide a comprehensive introduction to the methodologies and applications of nonparametric change-point detection. [17] construct a nonparametric minimax-optimal test to discriminate continuous paths with volatility jumps and prove weak convergence of the test statistic to an extreme value distribution. In the online setting, [5] present a meta-algorithm which compares data in some "reference window" to the data in the current window, using empirical distance measures that are not kernel-based; [7] detect abrupt changes by comparing two sets of descriptors extracted online from the signal at each time instant: the immediate past set and the immediate future set, and then use a soft margin single-class support vector machine to build a dissimilarity measure in the feature space between those sets without estimating densities as an intermediate step, which is asymptotically equivalent to the Fisher ratio in the Gaussian case; [6] present a density-ratio estimation method to detect change-points, fitting the density-ratio using a non-parametric Gaussian kernel model, whose parameters are updated online via stochastic gradient descent approach. Another important branch of nonparametric change-point detection method is based on Kolmogorov-Smirnov test, in [18, 19], which has been used in [20]. The generalization of Kolmogorov-Smirnov test from the univariate setting to the multi-dimensional setting is given by [21], which, however, is less convenient to use than the kernel-based statistic test.

Seminal works by [22] study kernel based U-statistic for change-point detection. They show that the statistic indexed by the assumed change-point location parameter τ, after proper standardization and rescaling of time and magnitude, converges in distribution to a Gaussian process under the null, and converges to a deterministic path in probability under the alternative distribution when the number of samples goes to infinity. These results are useful for bounding the detection statistics under the null with high-probability (hence, controlling the false detection), and for studying the consistency of tests. [23] and [24] contain comprehensive discussions on asymptotic theory of nonparametric statistics including

U-statistics. Our scan B-statistic can also be viewed as a form of U-statistic using an appropriate definition of the kernel. The main differences between these classic works from our proposed scan B-statistic are: (1) our statistic uses B-test block decomposition and averaging to make the test statistic more computationally efficient; (2) our statistic is more challenging to analyze due to the block structure and correlation introduced by scan statistics; (3) our analytical approach is different: [22] leverage invariance principle to establish convergence of the entire sample path; we focus on characterizing the tail probability of the statistic under the null and use the change-of-measure technique to achieve good approximation accuracy.

Other existing works that also focus on establishing asymptotic distribution of the detection statistic under the null for controlling the false alarm rate include the following: [2] present a maximum kernel Fisher discriminant ratio statistic and study its asymptotic null distribution; [25] investigate the two-sample test U-statistic for dependent data. Our approach is different from above in that we focus on directly approximating the tail of the detection statistic under the null, rather than trying to obtain its asymptotic distribution. Moreover, traditional analyses are usually done for offline change-point detection, while our analytical framework based on change-of-measure can be applied to both offline and online change-point detection.

Change-point detection problems are related to the classical statistical two-sample test. However, they are usually more challenging than the two-sample test because the change-point location τ is unknown. Hence, when forming the detection statistic, one has to "take the maximum" of the detection statistics. The statistics being maxed over are usually highly correlated since they are computed using overlapping data.

Our techniques for approximating false alarm rates differ from large-deviation techniques in [26], which establish exponential rate by which the probability converges to zero. In certain scenarios, the first-order approximation obtained from large-deviation techniques may not be sufficient for choosing threshold. Our method provides more refined approximations to include polynomial terms and constants.

Finally, there are also works taking different approaches rather than hypothesis test for change-point detection. For instance, [27] develop a kernel-based multiple-change-point detection approach, where the optimal location to segment the data is obtained by dynamic programming; [28] estimates multiple change-points by developing a kernelized linear model, and they provide a non-asymptotic oracle inequality for the estimation error. In the offline setting, [4] study a problem when there are s anomalous sequences out of n sequences to be detected, and the test statistic is constructed using MMD; [29] propose a nonparametric approach based on U-statistics and adopt the hierarchical clustering, which is capable of consistently estimating an unknown number of multiple change-point locations; [30] propose a nonparametric maximum likelihood approach, with the number of change-points determined from the Bayesian information criterion (BIC) and the locations of the change-points estimated via dynamic programming.

Our notations are standard. Let I_k denote the identity matrix of size k-by-k. Let $\mathbb{E}[\mathcal{A};\mathcal{B}] = \mathbb{E}[\mathcal{A}\mathbf{1}_{\mathcal{B}}]$ denote the expectation conditioned on event $\mathcal{B}$, where $\mathbf{1}_{\mathcal{B}}$ represents the indicator function that takes value 1 when the event $\mathcal{B}$ happens and takes value 0, otherwise. Let $\mathrm{Var}(\cdot)$ and $\mathrm{Cov}(\cdot)$ denote the variance and the covariance. Let $\mathbf{0}$ and $\mathbf{e}$ denote vectors of all zeros and all ones, respectively. Let $[\Sigma]_{ij}$ denote the ij-th element of a matrix Σ. In Section 3.5, $\mathbb{E}_B$, Var_B, and Cov_B denote the values computed under the new probability measure $\mathbb{P}_B$ after the change-of-measure, where B is the block size. Similarly, in Section 2.4.2, $\mathbb{E}_t$, Var_t, and Cov_t denote the values obtained under the new probability measure $\mathbb{P}_t$ after the change-of-measure, where t is the time index.

2.2 Background

We first briefly review the reproducing kernel Hilbert space (RKHS) and the *maximum mean discrepancy* (MMD). A RKHS $\mathcal{F}$ on $\mathcal{X}$ with a kernel $k(x, x')$ is a Hilbert space of functions $f(\cdot) : \mathcal{X} \mapsto \mathbb{R}$ equipped with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$. Its element $k(x, \cdot)$ satisfies the reproducing property: $\langle f(\cdot), k(x,\cdot)\rangle_{\mathcal{F}} = f(x)$, and consequently, $\langle k(x,\cdot), k(x',\cdot)\rangle_{\mathcal{F}} = k(x,x')$, meaning that we can view the evaluation of a function f at any point $x \in \mathcal{X}$ as an inner product. Commonly used RKHS kernel functions include the Gaussian radial basis function (RBF) $k(x,x') = \exp(-\|x-x'\|^2/2\sigma^2)$, where $\sigma > 0$ is the kernel bandwidth, and polynomial kernel $k(x,x') = (\langle x, x'\rangle + a)^d$, where $a > 0$ and $d \in \mathbb{N}$ (see [31]). RKHS kernels can also be defined for sequences, graph and other structured object (see [32]). In this paper, if not otherwise stated, we will assume that Gaussian RBF kernel is used.

Assume there are two sets X and Y, each with n samples taking value on a general domain $\mathcal{X}$, where $X = \{x_1, x_2, \ldots, x_n\}$ are *i.i.d.* with a distribution P, and $Y = \{y_1, y_2, \ldots, y_n\}$ are *i.i.d.* with a distribution Q. The MMD is defined as [9]

$$\text{MMD}[\mathcal{F}, P, Q] := \sup_{f\in\mathcal{F}} \left\{\mathbb{E}_{X\sim P}[f(X)] - \mathbb{E}_{Y\sim Q}[f(Y)]\right\}.$$

An unbiased estimator of MMD^2 can be obtained using U-statistic [9]

$$\text{MMD}_u^2[\mathcal{F}, X, Y] = \frac{1}{n(n-1)} \sum_{i\neq j}^{n} h(x_i, x_j, y_i, y_j), \tag{2.1}$$

where $h(\cdot)$ is the kernel for U-statistic and it can be defined using an RKHS kernel as

$$h(x_i, x_j, y_i, y_j) = k(x_i, x_j) + k(y_i, y_j) - k(x_i, y_j) - k(x_j, y_i). \tag{2.2}$$

Intuitively, the empirical test statistic MMD$_u^2$ is expected to be small (close to zero) if $P = Q$, and large if P and Q are "far" apart. The complexity for evaluating MMD$_u^2$ is $\mathcal{O}(n^2)$, since we have to form the so-called Gram matrix for the data, which is of size n-by-n. Under the null hypothesis, $P = Q$, the U-statistic is degenerate and has the same distribution as an infinite sum of Chi-square variables.

To improve computational efficiency, an alternative approach to eatimate MMD^2, called the B-test, is presented by [8]. The key idea is to partition the n samples from P and Q

into N non-overlapping blocks, $X_1, \dots, X_N$ and $Y_1, \dots, Y_N$, each of size B. Then one computes $\text{MMD}_u^2[\mathcal{F}, X_i, Y_i]$ for each pair of blocks and takes an average:

$$\text{MMD}_B^2[\mathcal{F}, X, Y] = \frac{1}{N} \sum_{i=1}^{N} \text{MMD}_u^2[\mathcal{F}, X_i, Y_i].$$

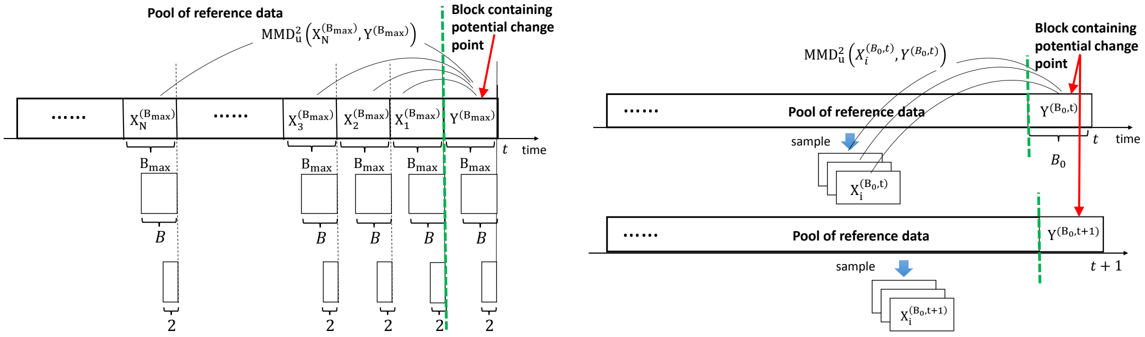
Since B is constant and N is on the order of $\mathcal{O}(n)$, the computational complexity of $\text{MMD}_B^2[\mathcal{F}, X, Y]$ is $\mathcal{O}(nB^2)$, which is significantly lower than the $\mathcal{O}(n^2)$ complexity of $\text{MMD}_u^2[\mathcal{F}, X, Y]$. Furthermore, by averaging $\text{MMD}_u^2[\mathcal{F}, X_i, Y_i]$ over blocks, when blocks are independent, the B-test statistic is asymptotically normal under the null using central limit theorem. This property allows a simple threshold to be derived for the B-test.

2.3 Scan B-statistics

Now we present our change-point detection procedure based on *scan B-statistic*. Consider a sequence of data $\{\dots, x_{-2}, x_{-1}, x_0, x_1, \dots, x_t\}$, each taking value on a general domain $\mathcal{X}$. Let $\{\dots, x_{-2}, x_{-1}, x_0\}$ denote the reference data that we know to follow a given pre-change distribution. Assume there is a large amount of reference data.

In offline change-point detection, the number of samples is fixed, and our goal is to detect the *existence* of a change-point τ, such that before the change-point, the samples are *i.i.d.* with a distribution P, and after the change-point, the samples are *i.i.d.* with a different distribution Q. The location τ where the change-point occurs is unknown. In other words, we are concerned with testing the null hypothesis

$$H_0 : x_i \sim P, \quad i = 1, \dots, t,$$



(a): offline (b): sequential

Figure 2.1: Illustration of offline change-point detection and online change-point detection.

against the single change-point alternative

$$H_1 : \exists 1 \leq \tau < t \quad x_i \sim \begin{cases} Q, & i > \tau \\ P, & \text{otherwise.} \end{cases}$$

Note that we are interested in the case of a sustained change: before the change, all samples follow one distribution, and after the change, all samples follow another distribution and never switch back. In online change-point detection, the number of samples is not fixed, and the goal is to detect the *emergence* of a change-point as quickly as possible. In various change-point detection settings, the number of post-change samples is small, but the number of reference samples is large. Therefore, when constructing MMD statistics over blocks, we will use a common post-change block and multiple disjoint pre-change reference blocks.

2.3.1 Offline change-point detection

For each possible change location τ, the post-change block consists of the most recent samples indexed from τ to t. Since we do not know the change-point location, we scan all possible change-point locations τ. This corresponds to considering a range of post-change block sizes B ranging from two (i.e., the most recent two samples are post-change samples) to $B_{\max}$. Here, we exclude $B = 1$ because the corresponding MMD is unable to compute.

The detection statistic is constructed as follows, also illustrated in Figure 3.3(a). Data are split into N reference blocks and one test block, each block is size of $B_{\max}$. Then we select data from each block to form smaller sub-blocks of various size B, $2 \leq B \leq B_{\max}$. The reference blocks are denoted as $X_i^{(B)}$, $i = 1, \ldots, N$, and the test block as $Y^{(B)}$. We compute MMD_u^2 for each reference sub-block with respect to the *common* post-change block, and take an average:

$$Z_B = \frac{1}{N} \sum_{i=1}^{N} \text{MMD}_u^2(X_i^{(B)}, Y^{(B)}). \tag{2.3}$$

Since the estimator MMD_u^2 is unbiased, under the null hypothesis $P = Q$, $\mathbb{E}[Z_B] = 0$. Let $\mathrm{Var}[Z_B]$ denote the variance of Z_B under the null. The variance of Z_B depends on the block size B and the number of blocks N. To have a fair comparison, we normalize each Z_B by their standard deviation

$$Z'_B = Z_B / (\text{Var}[Z_B])^{1/2},$$

and take the maximum over all B to form the *offline scan B-statistic*. The variance $\mathrm{Var}[Z_B]$ is given in Lemma 1. The closed-form expression facilitates the estimation of the variance of the statistic. A change-point is detected whenever the offline scan B-statistic exceeds a pre-specified threshold b:

$$\max_{2 \leq B \leq B_{\max}} Z'_B > b. \quad \{\text{offline change-point detection}\} \tag{2.4}$$

2.3.2 Online change-point detection

In the online setting, new samples sequentially and we constantly test whether the incoming samples come from a different distribution. To reduce computational burden, in the online setting, we fix the block-size and adopt a *sliding window* approach. The resulted sliding window procedure can be viewed as a type of Shewhart chart by [33].

The detection statistic is constructed as follows, also illustrated in Figure 3.3(b). At each time t, we treat the most recent B_0 samples as the post-change block. In online change-point detection, we want to detect the change as quickly as possible. Hence, typically we will not wait till collecting many post-change samples. On the other hand, there is a large amount of reference data. To utilize data efficiently, we utilize a common test block consisting of the most recent samples to form the statistic with N different reference blocks. The reference blocks are formed by taking NB_0 samples without replacement from the reference pool. We compute MMD_u^2 between each reference block with respect to the common post-change block, and take an average:

$$Z_{B_0,t} = \frac{1}{N}\sum_{i=1}^{N} \text{MMD}_u^2(X_i^{(B_0,t)}, Y^{(B_0,t)}), \tag{2.5}$$

where B_0 is the fixed block-size, $X_i^{(B_0,t)}$ is the i-th reference block at time t, and $Y^{(B_0,t)}$ is the the post-change block at time t. When there are new samples, we append them to the post-change block and purge the oldest samples. We show later that this construction allows for an explicit characterization of the false-alarm rate. We divide each statistic by its standard deviation to form the *online scan B-statistic*:

$$Z'_{B_0,t} = Z_{B_0,t}/(\text{Var}[Z_{B_0,t}])^{1/2}.$$

The calculation of $\mathrm{Var}[Z_{B_0,t}]$ can also be achieved using Lemma 1. The online change-point detection procedure is a stopping time: an alarm is raised whenever the detection statistic exceeds a pre-specified threshold $b > 0$:

$$T = \inf\{t : Z'_{B_0,t} > b\}. \quad \{\text{online change-point detection}\} \tag{2.6}$$

The online scan B-statistic can be computed efficiently. Note that the variance of the

$Z_{B_0,t}$ only depends on the block size B_0 but is independent of t. Hence, it can be pre-computed. Moreover, there is a simple way to compute the online B-statistic recursively, as specified in Appendix A.1.

2.3.3 Analytical expression for $\mathrm{Var}[Z_B]$

We obtain an analytical expression for $\mathrm{Var}[Z_B]$, which is useful when forming the detection statistic in (2.4) and (2.6).

Lemma 1 (Variance of Z_B under the null) *Given block size $B \geq 2$ and the number of blocks N, under the null hypothesis,*

$$\mathrm{Var}[Z_B] = \binom{B}{2}^{-1} \left(\frac{1}{N} \mathbb{E}[h^2(x, x', y, y')] + \frac{N-1}{N} \mathrm{Cov}\left[h(x, x', y, y'), h(x'', x''', y, y')\right] \right), \tag{2.7}$$

where x, x', x'', x''', y, and y' are i.i.d. random variables with the null distribution P.

The lemma is proved by making a connection between MMD_u^2 and U-statistic in [24] and utilizing the properties of U-statistic. A detailed proof is provided in Appendix A.2.

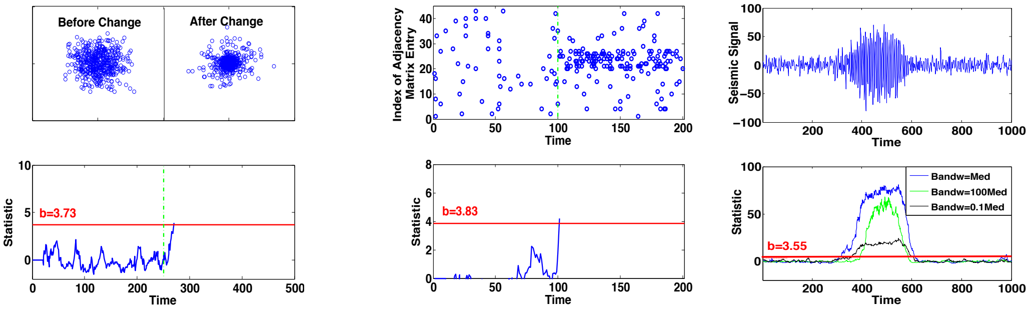
2.3.4 Examples of detection statistics

Below, we present a few examples to demonstrate that the B-statistics is quite robust in various settings with different distributions.

Gaussian to Gaussian mixture. In Figure 2.2(a), $P = \mathcal{N}(0, I_2)$, Q is a mixture Gaussians: $0.3\mathcal{N}(0, I_2) + 0.7\mathcal{N}(0, 0.1I_2)$, and $\tau = 250$. The online procedure stops at time 270 meaning the change is detected with a small delay of 20 unit time.

Sequence of graphs. In Figure 2.2(b), we consider detecting the emergence of a community inside a network, which modeled using a stochastic block model, as considered by [34]. Assume that before the change, each sample is a realization of an Erdős-Rényi random graph, with the probability of forming an edge $p_0 = 0.1$ uniformly across the graph.

After the change, a “community” emerges, which is a subset of nodes, where the edges are formed in between these nodes with much higher probability $p_1 = 0.3$. The post-change distribution models a community where the members of the community interact more often. Our online procedure stops at time 102, meaning the change is detected with a small delay of 2 unit times.



(a): Gaussian to GMM, $\tau = 250$ (b) Graphs, $\tau = 100$ (c): Real seismic signal

Figure 2.2: Examples of scan B-statistics.

Real seismic signal and effect of kernel bandwidth. In Figure 2.2(c), we consider a segment of real seismic signal that contains a change-point. Using the seismic signal, we illustrate the effect of different kernel bandwidth. For Gaussian RBF kernel $k(Y, Y') = \exp\left(-\|Y - Y'\|^2/2\sigma^2\right)$, the kernel bandwidth $\sigma > 0$ is typically chosen using a “median trick” in [31, 35], where σ is set to be the median of the pairwise distances between data points.

2.4 Theoretical approximations

2.4.1 Theoretical approximation for significance level of offline scan B-statistic

In the offline setting, the choice of the threshold b involves a tradeoff between two standard performance metrics: (1) significance level (SL), which is the probability that the statistic exceeds the threshold b when the null hypothesis is true (i.e., when there is no change); and (2) power, which is the probability of the statistic exceeds the threshold when the alternative hypothesis is true.

We present an accurate approximation to the SL of the offline scan B-statistic, assuming the detection threshold b tends to infinity and the number of blocks N is fixed. The following theorem is our main result.

Theorem 2 (SL of offline scan B-statistic) *When $b \to \infty$, and $B_{\max} \to \infty$, with $b/(B_{\max})^{1/2}$ held as a fixed positive constant, the significance level of the offline B-statistic defined in (2.4) is given by*

$$\mathbb{P}\left\{\max_{2\leq B\leq B_{\max}} Z'_B > b\right\} = be^{-\frac{1}{2}b^2} \cdot \sum_{B=2}^{B_{\max}} \frac{(2B-1)}{2\sqrt{2\pi}B(B-1)}\nu\left(b\sqrt{\frac{2B-1}{B(B-1)}}\right) \cdot [1+o(1)]\,, \tag{2.8}$$

where the special function

$$\nu(\mu) \approx \frac{(2/\mu)(\Phi(\mu/2)-0.5)}{(\mu/2)\Phi(\mu/2)+\phi(\mu/2)}, \tag{2.9}$$

$\phi(x)$ and $\Phi(x)$ are the probability density function and the cumulative distribution function of the standard normal distribution, respectively.

Although the approximation (2.8) is derived in the asymptotic regime and under the assumption that the collection of random variables $\{Z'_B\}_{B=2,\ldots,B_{\max}}$ form a Gaussian random field, we can show numerically that (2.8) is quite accurate in the non-asymptotic regime. Consider synthetic data that are *i.i.d.* normal $P = \mathcal{N}(0, I_{20})$. We set $B_{\max}$ to be 50, 100, 150, and in each case, $N = 5$. We compare the thresholds obtained by (2.8) and by simulation, for a prescribed SL α. To obtain threshold by simulation, we generate Monte Carlo trials for offline B-statistics and find the $(1-\alpha)$-quantile as the estimated threshold. Table 2.1 shows that for various choices of $B_{\max}$, the thresholds predicted by Theorem 2 match quite well with those obtained by simulation. The accuracy can be further improved for smaller α values by skewness correction as shown in Section 2.6.

The complete proof of Theorem 2 can be found in Appendix A.3, which leverages

Table 2.1: Thresholds for the offline scan B-statistics obtained by simulation, theory (Theorem 2), and theory with Skewness Correction (Section 2.6).

α	$B_{\max} = 50$			$B_{\max} = 100$			$B_{\max} = 150$		
	b (sim)	b (theory)	b (SC)	b (sim)	b (theory)	b (SC)	b (sim)	b (theory)	b (SC)
0.10	2.41	**2.38**	2.57	2.43	**2.50**	2.76	2.53	**2.56**	2.89
0.05	2.77	**2.67**	2.97	2.76	**2.78**	3.17	2.97	**2.83**	3.22
0.01	3.54	3.23	**3.64**	3.47	**3.32**	3.82	3.64	3.37	**3.89**

the change-of-measure technique. In a nutshell, we aim to find the probability of a rare event: under null the distribution, the boundary exceeding event $\{\max_{2 \le B \le B_{\max}} Z'_B > b\}$ for a large threshold b is rare (so that false alarm remains low). Since quantifying such a small probability is hard under the null distribution, we consider an alternative probability measure under which this boundary exceeding event happens with much higher probability. Under the new measure, one can use the local central limit theorem to a obtain an analytical expression for the probability. In the end, the original small probability will be related to the probability under the alternative measure using the Mill's ratio in [11].

The proof assumes the collection of random variables $\{Z'_B\}_{B=2,\ldots,B_{\max}}$ form a Gaussian random field (as an approximation). This means the finite-dimensional joint distributions of the collection of random variables are all Gaussian, and they are completely specified by the mean and the covariance functions, which we characterize below (this is useful for establishing Theorem 2). These results will be used when we quantify the tail probability of the scan B-statistics. Under the null distribution, the expectation $\mathbb{E}[Z'_B]$ is zero due to the unbiased property of the MMD estimator. The covariance under the null distribution is given by the following lemma:

Lemma 3 (Covariance structure of Z'_B in the offline setting) *Under the null distribution, the covariance of* $\{Z'_B\}_{B=2,\ldots,B_{\max}}$ *is given by*

$$r_{u,v} = \mathrm{Cov}\left(Z'_u, Z'_v\right) = \sqrt{\binom{u}{2}\binom{v}{2}} \Big/ \binom{u \vee v}{2}, \quad 2 \le u, v \le B_{\max}, \tag{2.10}$$

where $u \vee v = \max\{u, v\}$.

The proof can be found in Appendix A.2.2.

2.4.2 Theoretical approximation for ARL of online scan B-statistic

In the online setting, two commonly used performance metrics are (see, e.g., [36]): (1) the average run length (ARL), which is the expected time before incorrectly announcing a change of distribution when none has occurred; (2) the expected detection delay (EDD), which is the expected time to fire an alarm when a change occurs immediately at $\tau = 0$. The EDD considers the worst case and provides an upper bound on the expected delay to detect a change-point when the change occurs later in the sequence of observations.

We present an accurate approximation to the ARL of online scan B-statistics. The approximation is quite useful in setting the threshold. As a result, given a target ARL, one can determine the corresponding threshold value b from the analytical approximation, avoiding the more expensive numerical simulations. Our main result is the following theorem.

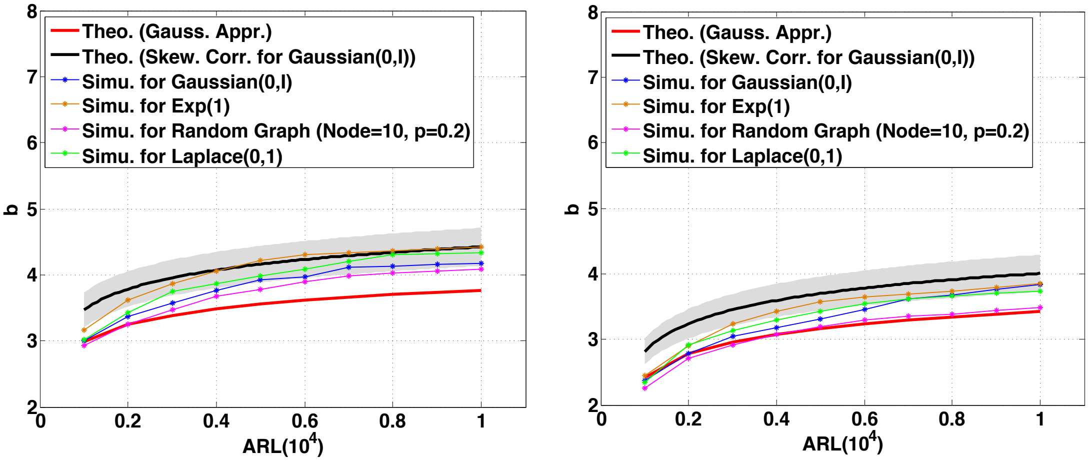
Theorem 4 (ARL in online scan B-statistic) *Let $B_0 \geq 2$. When $b \to \infty$, the ARL of the stopping time T defined in (2.6) is given by*

$$\mathbb{E}[T] = \frac{e^{b^2/2}}{b} \cdot \left\{ \frac{(2B_0 - 1)}{\sqrt{2\pi} B_0 (B_0 - 1)} \cdot \nu \left(b \sqrt{\frac{2(2B_0 - 1)}{B_0(B_0 - 1)}} \right) \right\}^{-1} \cdot [1 + o(1)]\,. \tag{2.11}$$

The complete proof of Theorem 11 is given in Appendix A.4.

We verify the accuracy of the approximation numerically, by comparing the thresholds obtained by Theorem 11 with those obtained from Monte Carlo simulation. Consider several cases of null distributions: standard normal $\mathcal{N}(0, 1)$, exponential distribution with mean 1, Erdős-Rényi random graph with ten nodes and probability of 0.2 of forming random edges, as well as Laplace distribution with zero mean and unit variance. The simulation results are obtained from 5000 direct Monte Carlo trials. As shown in Figure 2.3, the thresholds predicted by Theorem 11 are quite accurate. Figure 2.3 also demonstrated that theory is quite accurate for various block sizes (especially for larger B_0). However, we also note that theory tends to underestimate the thresholds. This is especially pronounced for small B_0, e.g., $B_0 = 50$. The accuracy of the theoretical results can be improved by skewness correction, shown by black lines in Figure 2.3, which are discussed later in Section 2.6.

Theorem 11 shows that ARL is $\mathcal{O}(e^{b^2})$ and, hence, b is $\mathcal{O}((\log \mathrm{ARL})^{1/2})$. Note that EDD is typically on the order of b/Δ due to Wald's identity [12], where Δ is the Kullback-Leibler (KL) divergence between the null and the alternative distributions (a constant). Hence, given the desired ARL (typically on the order of 5000 or 10000), the error in the estimated threshold will only be translated linearly to EDD. This is a blessing since it means typically a reasonably accurate b will cause little performance loss in EDD. Similarly, Theorem 2 shows that SL is $\mathcal{O}(e^{-b^2})$ and a similar argument can be made for the offline case.



(a): $B_0 = 50$ (b): $B_0 = 200$

Figure 2.3: Comparison of ARL obtained from simulation, from Theorem 11, and with the skewness correction (Section 2.6).

2.5 Detection power study

In this section, we study the detection power and the expected detection delay of the offline and online scan B-statistics, respectively, and compare them with classic methods.

2.5.1 Offline change-point detection: Comparison with parametric statistics

We compare the offline scan B-statistic with two commonly used parametric test statistics: the Hotelling's T^2 and the generalized likelihood ratio (GLR) statistics. Assume samples $\{x_1, x_2, \ldots, x_n\}$.

Hotelling's T^2 statistic. For a hypothetical change-point location τ, we can define the Hotelling's T^2 statistic for samples in two segments $[1, \tau]$ and $[\tau + 1, t]$ as

$$T^2(\tau) = \frac{\tau(n-\tau)}{n}(\bar{x}_\tau - \bar{x}^*_\tau)^T \widehat{\Sigma}^{-1}(\bar{x}_\tau - \bar{x}^*_\tau),$$

where, $\bar{x}_\tau = \sum_{i=1}^{\tau} x_i/\tau$, $\bar{x}^*_\tau = \sum_{i=\tau+1}^{n} x_i/(n-\tau)$ and the pooled covariance estimator

$$\widehat{\Sigma} = (n-2)^{-1}\left(\sum_{i=1}^{\tau}(x_i - \bar{x}_i)(x_i - \bar{x}_i)^T + \sum_{i=\tau+1}^{n}(x_i - \bar{x}^*_i)(x_i - \bar{x}^*_i)^T\right).$$

The Hotelling's T^2 test detects a change whenever $\max_{1\leq\tau\leq n} \max T^2(\tau)$ exceeds a threshold.

The **generalized likelihood ratio (GLR)** statistic can be derived by assuming the null and the alternative distributions are two multivariate normal distributions, and both the mean and the covariance matrix are all unknown. For a hypothetical change-point location τ, the GLR statistic is given by

$$\ell(\tau) = n\text{log}|\widehat{\Sigma}_n| - \tau\text{log}|\widehat{\Sigma}_\tau| - (n-\tau)\text{log}|\widehat{\Sigma}^*_\tau|,$$

where $\widehat{\Sigma}_\tau = \tau^{-1}\left(\sum_{i=1}^{\tau}(x_i - \bar{x}_i)(x_i - \bar{x}_i)^T\right)$, and $\widehat{\Sigma}^*_\tau = (n-\tau)^{-1}\sum_{i=\tau+1}^{n}(x_i - \bar{x}^*_i)(x_i - \bar{x}^*_i)^T$. The GLR statistic detects a change whenever $\max_{1\leq\tau\leq n} \ell(\tau)$ exceeds a threshold.

For our examples, we set $n = B_{\max} = 200$ for the Hotelling's T^2 and the scan B-statistics, respectively. Let the change-point occurs at $\tau = 100$, and choose the significance level $\alpha = 0.05$. The thresholds for the offline scan B-statistic are obtained from Theorem 2,

Table 2.2: Comparison of detection power for offline change-point detection.

	Case 1	Case 2	Case 3	Case 4
B-statistic	**0.71**	**1.00**	**1.00**	**0.44**
Hotelling's T^2	0.18	0.88	0.87	0.03
GLR	0.03	0.05	0.12	0.04

and those for the other two methods the thresholds are obtained from simulations. Consider the following cases:

Case 1 (mean shift): observe a sequence of observations in $\mathbb{R}^{20}$, whose distribution shifts from $\mathcal{N}(\mathbf{0}, I_{20})$ to $\mathcal{N}(0.1\mathbf{e}, I_{20})$;

Case 2 (mean shift with larger magnitude): observe a sequence of observations in $\mathbb{R}^{20}$, whose distribution shifts from $\mathcal{N}(\mathbf{0}, I_{20})$ to $\mathcal{N}(0.2\mathbf{e}, I_{20})$;

Case 3 (mean and local covariance change): observe a sequence of observations in $\mathbb{R}^{20}$, whose distribution shifts from $\mathcal{N}(\mathbf{e}, I_{20})$ to $\mathcal{N}(0.2\mathbf{e}, \Sigma)$, where $[\Sigma]_{11} = 2$ and $[\Sigma]_{ii} = 1$, $i = 2, \ldots, 20$;

Case 4 (Gaussian to Laplace): observe a sequence of one-dimensional observations, whose distribution shifts from $\mathcal{N}(0, 1)$ to Laplace distribution with zero mean and unit variance. Note that the mean and the variance remain the same after the change.

We estimate the power for each case using 100 Monte Carlo trials. Table 2.2 shows that the scan B-statistic achieves higher power than the Hotelling's T^2 statistic as well as the GLR statistic in all cases. The GLR statistic performs poorly, since when τ is small or closer to the end point, it estimates the pre-change and post-change sample covariance matrix using a very limited number of samples.

2.5.2 Online change-point detection: Comparison with Hotelling's T^2 statistics

Now consider the online scan B-statistic with a fixed block-size $B_0 = 20$. We compare the online scan B-statistic with a Shewhart chart based on Hotelling's T^2 statistic[1]. At

[1] Here we made no comparison of the online scan B-statistic with the GLR statistic, since in our experiments, Hotelling's T^2 consistently outperforms GLR when the dimension is high.

each time t, we form a Hotelling's T^2 statistic using the immediately past B_0 samples in $[t - B_0 + 1, t]$,

$$T^2(t) = B_0(\bar{x}_t - \hat{\mu})^T \widehat{\Sigma}_0^{-1}(\bar{x}_t - \hat{\mu}_0),$$

where $\bar{x}_t = (\sum_{i=t-B_0+1}^{t} x_i)/B_0$, and $\hat{\mu}_0$ and $\widehat{\Sigma}_0$ are estimated from reference data. The procedure detects a change-point whenever $T^2(t)$ exceeds a threshold for the first time. The threshold for online scan B-statistic is obtained from Theorem 11, and from simulations for the Hotelling's T^2 statistic. To simulate EDD, let the change occur at the first point of the testing data. Consider the following cases:

Case 1 (mean shift): distribution shifts from $\mathcal{N}(\mathbf{0}, I_{20})$ to $\mathcal{N}(0.3\mathbf{1}, I_{20})$;

Case 2 (covariance change): distribution shifts from $\mathcal{N}(\mathbf{0}, I_{20})$ to $\mathcal{N}(\mathbf{0}, \Sigma)$, where $[\Sigma]_{ii} = 2, i = 1, 2, \ldots, 5$ and $[\Sigma]_{ii} = 1,\ i = 6, \ldots, 20$;

Case 3 (covariance change): distribution shifts from $\mathcal{N}(\mathbf{0}, I_{20})$ to $\mathcal{N}(\mathbf{0}, 2I_{20})$;

Case 4 (Gaussian to Gaussian mixture): distribution shifts from $\mathcal{N}(\mathbf{0}, I_{20})$ to mixture Gaussian $0.3\mathcal{N}(\mathbf{0}, I_{20}) + 0.7\mathcal{N}(\mathbf{0}, 0.1I_{20})$;

Case 5 (Gaussian to Laplace)[2]: distribution shifts from $\mathcal{N}(0, 1)$ to Laplace distribution with zero mean and unit variance.

We evaluate the EDD for each case using 500 Monte Carlo trials. The results are summarized in Table 3.2. Note that in detecting changes in either Gaussian mean or covariance, the online scan B-statistic performs competitively with Hotelling's T^2, which is tailored to the Gaussian distribution. In the more challenging scenarios such as Case 4 and Case 5, the Hotelling's T^2 fails to detect the change-point whereas the online scan B-statistic can detect the change fairly quickly.

[2] For these difficult situations, we report the EDD comparisons based on the selected 500 sequences where B-statistics successfully detect the changes, which are defined as crossing the threshold within 50 steps from the time that the change occurs. Hotelling's T^2 fails to detect the changes for all sequences.

Table 2.3: Comparison of EDD in online change-point detection.

	Case 1	Case 2	Case 3	Case 4	Case 5
B-statistic	4.20	**9.10**	**1.00**	**23.38**	**23.03**
Hotelling's T^2	**2.47**	25.46	1.27	–	–

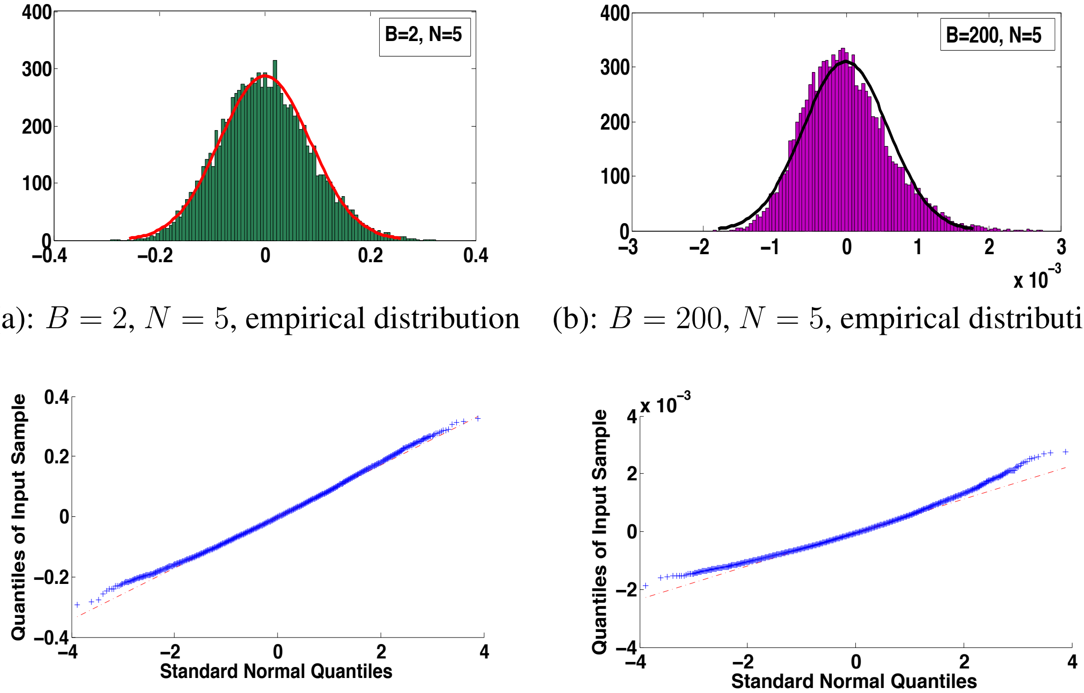
2.6 Skewness correction

We have shown that approximations to the significance level and ARL, assuming that random variables $\{Z'_B\}_{B=2,3,\ldots}$ form a Gaussian random field, are reasonably accurate. However, Z'_B does not converge to normal distribution even when B is large (see Appendix A.6) and it has a non-vanishing skewness, as illustrated by the following numerical example. Form 10000 instances of Z_B computed using samples from $\mathcal{N}(0, I_{20})$. Figures 2.4(a)-(b) show the empirical distributions of Z_B when $N = 5$, and $B = 2$ or $B = 200$, respectively. Also plotted are the Gaussian probability density functions with mean equal to the sample mean, and the variance predicted by Lemma 1. Note that the empirical distributions of Z_B match with Gaussian distributions to a certain extent but the skewness becomes larger for larger B. Figures 2.4(c)-(d) show the corresponding Q-Q plots.

To incorporate the skewness of Z_B, one can improve the accuracy of the approximations for significance level in Theorem 2 and for ARL in Theorem 11. Note that the log moment generating function $\psi(\theta)$ defined in (A.6) corresponds to the cumulant generating function [37] and it has an expansion for θ close to zero:

$$\psi(\theta) = \kappa_1\theta + \frac{\kappa_2}{2}\theta^2 + \frac{\kappa_3}{3!}\theta^3 + o(\theta^3).$$

Since $\mathbb{E}[Z'_B] = 0$, the cumulants take values $\kappa_1 = \mathbb{E}[Z'_B] = 0$, $\kappa_2 = \mathrm{Var}[Z'_B] = 1$, $\kappa_3 = \mathbb{E}[(Z'_B)^3] - 3\mathbb{E}[(Z'_B)^2]\mathbb{E}[Z'_B] + 2(\mathbb{E}[Z'_B])^3 = \mathbb{E}[(Z'_B)^3]$. Recall that when deriving approximations using change-of-measurement, we choose parameter θ such that $\dot{\psi}(\theta) = b$. If Z'_B is a standard normal, $\psi(\theta) = \theta^2/2$, and hence $\theta = b$. Now with skewness correction,



(a): $B = 2$, $N = 5$, empirical distribution (b): $B = 200$, $N = 5$, empirical distribution

(c): $B = 2$, $N = 5$, Q-Q plot (d): $B = 200$, $N = 5$, Q-Q plot

Figure 2.4: Empirical distributions of Z_B when $B = 2$ and $B = 200$.

we approximate $\psi(\theta)$ as $\theta^2/2 + \kappa_3\theta^3/6$ when solving for θ. Hence, we solve for

$$\dot{\psi}(\theta) \approx \theta + \mathbb{E}[(Z'_B)^3]\theta^2/2 = b,$$

and denote the solution to be θ_B (note that this time the solution depends on B). Moreover, with skewness correction, we will change the leading exponent term in (2.8) and (A.28) from $e^{-b^2/2}$ to be $e^{\psi(\theta'_B)-\theta'_B b}$.

From numerical experiments, we find that the skewness correction is especially useful when the significance level is small (e.g., $\alpha = 0.01$) for the offline case, when block size B_0 is small (see Table 2.1 and Fig. 2.3), and can be important for real data where the data are noisy and the null distribution is more difficult to characterize.

For example, we consider real speech data from the CENSREC-1-C dataset (more details in Section 2.7). Here, the null distribution P corresponds to the unknown distribution of the background signal, and we are interested in detecting the onset of speech signals. This case is more challenging because the true distribution can be arbitrary. In the dataset, there are 3000 reference samples. We bootstrap these reference samples to generate 10000 re-samples to estimate the tail of the detection statistic. Table 2.4 demonstrates that the thresholds predicted by the expensive bootstrapping, by Theorem 2), and by theory with skewness correction, respectively, for various SL values α. Note that in this case, the accuracy improves significantly by skewness correction.

Table 2.4: Thresholds for the offline scan B-statistics using speech data, obtained by simulation, theory (Theorem 2), and theory with skewness correction.

α	$B_{\max} = 50$			$B_{\max} = 100$			$B_{\max} = 150$		
	b (boot)	b (theory)	b (SC)	b (boot)	b (theory)	b (SC)	b (boot)	b (theory)	b (SC)
0.10	2.96	2.38	**3.23**	3.16	2.50	**3.59**	3.21	**2.56**	3.94
0.05	3.62	2.67	**3.68**	3.82	2.78	**4.06**	3.86	2.83	**4.43**
0.01	4.85	3.23	**4.61**	5.20	3.32	**5.03**	5.42	3.37	**5.45**

The remaining task is to estimate the skewness of scan B-statistic. Since Z_B is zero-mean, the skewness of Z'_B is related to the variance and third moment of Z_B via

$$\kappa_3 = \mathbb{E}[(Z'_B)^3] = \mathrm{Var}[Z_B]^{-3/2}\mathbb{E}[Z_B^3].$$

We already know how to estimate the variance of Z_B from Lemma 1. The following lemma shows the third-order moment $\mathbb{E}[Z_B^3]$ in terms of the moments of the kernel h defined in (2.2):

Lemma 5 (Third-order moment of Z_B)

$$
\begin{aligned}
\mathbb{E}[Z_B^3] = \frac{8(B-2)}{B^2(B-1)^2} & \left\{ \frac{1}{N^2}\mathbb{E}\left[h(x,x',y,y')h(x',x'',y',y'')h(x'',x,y'',y)\right] \right. \\
& + \frac{3(N-1)}{N^2}\mathbb{E}\left[h(x,x',y,y')h(x',x'',y',y'')h(x''',x'''',y'',y)\right] \\
& \left. + \frac{(N-1)(N-2)}{N^2}\mathbb{E}\left[h(x,x',y,y')h(x'',x''',y',y'')h(x'''',x''''',y'',y)\right] \right\} \\
+ \frac{4}{B^2(B-1)^2} & \left\{ \frac{1}{N^2}\mathbb{E}\left[h(x,x',y,y')^3\right] \right. \\
& + \frac{3(N-1)}{N^2}\mathbb{E}\left[h(x,x',y,y')^2 h(x'',x''',y,y')\right] \\
& \left. + \frac{(N-1)(N-2)}{N^2}\mathbb{E}\left[h(x,x',y,y')h(x'',x''',y,y')h(x'''',x''''',y,y')\right] \right\}.
\end{aligned}
\tag{2.12}
$$

The proof can be found in Appendix A.5. Lemma 5 enables us to estimate the skewness efficiently, by reducing it to evaluating simpler terms in (2.12) that only requires estimating the statistic of the kernel function $h(\cdot,\cdot,\cdot,\cdot)$ with tuples of samples.

Finally, although Z_B' does not converge to Gaussian, the difference between its moment generating functions and that of the standard normal distribution can be bounded, as we show below. By applying an argument on Page 220 of [11], we obtain that

$$\left|\mathbb{E}[e^{\theta Z_B'}] - (1+\frac{\theta^2}{2})\right| \le \min\{\frac{|\theta|^3}{6}\mathbb{E}[|Z_B'|^3], \theta^2\mathbb{E}[|Z_B'|^2]\}.$$

If considering the skewness κ_3 of Z_B', we have a better estimation

$$\left|\mathbb{E}[e^{\theta Z_B'}] - (1+\frac{\theta^2}{2}+\frac{\theta^3\kappa_3}{6})\right| \le \min\{\frac{\theta^4}{24}\mathbb{E}[|Z_B'|^4], \frac{1}{3}|\theta|^3\mathbb{E}[|Z_B'|^3]\}.$$

2.7 Real data

We test the performance of the scan B-statistics for change-point detection on real data. Our datasets include: (1) CENSREC-1-C: a real-world speech data set in the Speech Resource Consortium (SRC) corpora provided by National Institute of Informatics (NII)[3]; (2) Human Activity Sensing Consortium (HASC) challenge 2011 data[4]. We compare our proposed scan B-statistics with a baseline algorithm, the relative density-ratio (RDR) estimate [6]. One limitation of the RDR algorithm, however, is that it is not suitable for high-dimensional data because estimating density ratio in the high-dimensional setting is an ill-posed problem. To achieve reasonable performance for the RDR algorithm, we adjust the bandwidth and the regularization parameter at each time step and, hence, the RDR algorithm is computationally more expensive than using the scan B-statistics. We adopt the standard Area Under Curve (AUC) as in [6] for our performance metric. The larger the AUC, the better.

Our scan B-statistics demonstrate competitive performance compared with the baseline RDR algorithm on the real data. Here we only report the main results and leave the details in Appendix A.7. For speech data, our goal is to online detect the emergence of a speech signal from the background. The backgrounds are taken from real acoustic signals, such as noise recorded in highway, airport and subway stations. The overall AUC for the scan B-statistic is **0.8014** and for the baseline algorithm is **0.7578**. For human activity detection data, our goal is to detect a transition from one activity to another as quickly as possible. Each instance consists of six possible human activity signals collected by portable three-axis accelerometers. The overall AUC for the scan B-statistic is **0.8871** and for the baseline algorithm is **0.7161**.

2.8 Discussion

There are a few possible directions to extend our work. (1) Thus far, we have assumed that data are *i.i.d.* from a null distribution P and when the change happens, data are *i.i.d.* from an alternative distribution Q. Under these assumptions, we have developed the offline and online change-point detection algorithms based on the two-sample nonparametric test

[3] Available from http://research.nii.ac.jp/src/en/CENSREC-1-C.html

[4] Available from http://hasc.jp/hc2011

statistic MMD. One may relax the temporal independence assumption and extend scan B-statistics for dependent data by incorporating ideas from [38]. (2) We have demonstrated how the number of blocks and block size affect the performance of scan B-statistics. One can also explore how kernel bandwidth, as well as the dimensionality of the data, would affect the performance. An empirical observation is that the performance of MMD statistic degrades with the increasing dimensions of data. Some recent results for the kernel-based test can be found in [35]. We may adopt the idea of [35] to extend our scan B-statistics for detecting a change in high dimensions. (3) For an exceedingly high dimensional data set with large Gram matrix, one can perform random subsampling to reduce complexity similar to [39].

CHAPTER 3
DETECTING CHANGES IN DYNAMIC EVENTS OVER NETWORKS.

Large volume of networked streaming event data are becoming increasingly available in a wide variety of applications, such as social network analysis, Internet traffic monitoring and healthcare analytics. Streaming event data are discrete observation occurred in continuous time, and the precise time interval between two events carries a great deal of information about the dynamics of the underlying systems.

How to promptly detect changes in these dynamic systems using these streaming event data? In this chapter, we propose a novel change-point detection framework for multi-dimensional event data over networks. We cast the problem into sequential hypothesis test, and derive the likelihood ratios for point processes, which are computed efficiently via an EM-like algorithm that is parameter-free and can be computed in a distributed fashion. We derive a highly accurate theoretical characterization of the false-alarm-rate, and show that it can achieve weak signal detection by aggregating local statistics over time and networks. Finally, we demonstrate the good performance of our algorithm on numerical examples and real-world datasets from twitter and Memetracker.

3.1 Overview

Networks have become a convenient tool for people to efficiently disseminate, exchange and search for information. Recent attacks on very popular web sites such as Yahoo and eBay [40], leading to a disruption of services to users, have triggered an increasing interest in network anomaly detection. In the positive side, surge of hot topics and breaking news can provide business opportunities. Therefore, *early detection* of changes, such as anomalies, epidemic outbreaks, hot topics, or new trends among streams of data from networked entities is a very important task and has been attracting significant interests [40, 41, 42].

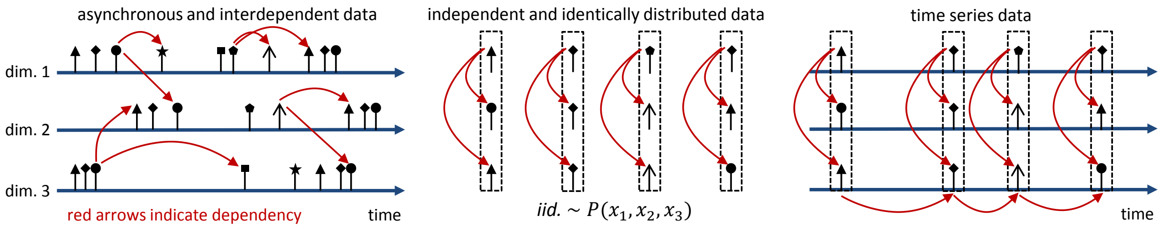

asynchronous and interdependent data
independent and identically distributed data
time series data
dim. 1
dim. 2
dim. 3
red arrows indicate dependency
time
iid. ~ $P(x_1, x_2, x_3)$
time

Figure 3.1: Asynchronously and interdependently generated high dimensional event data are fundamentally different from *i.i.d.* and time-series data.

All types of the above-mentioned changes can be more concretely formulated as the changes of time interval distributions between events, combined with the alteration of interaction structures across components in networks. However, change-point detection based on event data occurring over the network topology is nontrivial. Apart from the possible temporal dependency of the event data as well as the complex cross-dimensional dependence among components in network, event data from networked entities are usually not synchronized in time. Dynamic in nature, many of the collected data are discrete events observed irregularly in continuous time [43, 44]. The precise time interval between two events is random and carries a great deal of information about the dynamics of the underlying systems. These characteristics make such event data fundamentally different from independently and identically distributed (*i.i.d.*) data, and time-series data where time and space is treated as an index rather than random variables (see Figure 3.1 for further illustrations of the distinctive nature of event data vs. *i.i.d.* and time series data). Clearly, *i.i.d.* assumption can not capture temporal dependency between data points, while time-series models require us to discretize the time axis and aggregate the observed events into bins (such as the approach in [45] for neural spike train change detection). If this approach is taken, it is not clear how one can choose the size of the bin and how to best deal with the case when there is no event within a bin.

Besides the distinctive temporal and spatial aspect, there are three additional challenges using event data over network: (*i*) how to detect weak changes; (*ii*) how to update the statistics efficiently online; and (*iii*) how to provide theoretical characterization of the false-alarm-rate for the statistics. For the first challenge, many existing approaches usually use random or ad-hoc aggregations which may not pool data efficiently or lose statistical power to detect weak signals. Occurrence of change-points (e.g., epidemic outbreaks, hot topics, etc.) over networks usually evince a certain clustering behavior over dimensions and tend to synchronize in time. Smart aggregation over dimensions and time horizon would manifest the strength of signals and detect the change quicker [46]. For the second challenge, many existing change-point detection methods based on likelihood ratio statistics do not take into account computational complexity nor can be computed in a distributed fashion and, hence, are not scalable to large networks. Temporal events can arrive at social platforms in very high volume and velocity. For instance, every day, on average, around 500 million tweets are tweeted on Twitter [47]. There is a great need for developing efficient algorithms for updating the detection statistics online. For the third challenge, it is usually very hard to control false-alarms for change-point detection statistics over a large network. When applied to real network data, traditional detection approaches usually have a high false alarms [40]. This would lead to a huge waste of resources since every time a change-point is declared, subsequent diagnoses are needed. Lacking accurate theoretical characterization of false-alarms, existing approaches usually have to incur expensive Monte Carlo simulations to determine the false-alarms and are prohibitive for large networks.

Our contributions. In this chapter, we present a novel online change-point detection framework tailored to multi-dimensional intertwined event data streams over networks (or conceptual networks) tackling the above challenges. We formulate the problem by leveraging the mathematical framework of sequential hypothesis testing and point processes modeling, where before the change the event stream follows one point process, and after the change the event stream becomes a different point process. Our goal is to detect such changes *as quickly as possible* after the occurrences. We derive generalized likelihood ratio statistics, and present an efficient EM-like algorithm to compute the statistic online with streaming data. The EM-like algorithm is parameter-free and can be implemented in a distributed fashion and, hence, it is suitable for large networks.

Specifically, our contributions include the following:

(*i*) We present a new sequential hypothesis test and likelihood ratio approach for detecting changes for the event data streams *over networks*. We will either use the Poisson process as the null distribution to detect the appearance of temporal independence, or use the Hawkes process as the null distribution to detect the possible alteration of the dependency structure. For (inhomogeneous) Poisson process, time intervals between events are assumed to be independent and exponentially distributed. For Hawkes process, the occurrence intensity of events depends on the events that have occurred, which implies that the time intervals between events would be correlated. Therefore, Hawkes process can be thought of as a special autoregressive process in time, and multivariate Hawkes process also provides a flexible model to capture cross-dimension dependency in addition to temporal dependency. Our model explicitly captures the information diffusion (and dependencies) both over networks and time, and allows us to aggregate information for weak signal detection. Our proposed detection framework is quite general and can be easily adapted to other point processes.

In contrast, existing work on change-point detection for point processes has also been focused on a single stream rather than the multidimensional case with networks. These work including detecting change in the intensity of a Poisson process [48, 49, 50] and the coefficient of continuous diffusion process [51]; detecting change using the self-exciting Hawkes processes include trend detection in social networks [52]; detecting for Poisson processes using a score statistic [53].

(*ii*) We present an efficient expectation-maximization (EM) like algorithm for updating the likelihood-ratio detection statistic online. The algorithm can be implemented in a *distributed* fashion due to its structure: only neighboring nodes need to exchange information for the E-step and M-step.

(*iii*) We also present accurate theoretical approximation to the false-alarm-rate (formally the average-run-length or ARL) of the detection algorithm, via the recently developed change-of-measure approach to handle highly correlated statistics. Our theoretical approximation can be used to determine the threshold in the algorithm accurately.

(*iv*) Finally, we demonstrate the performance gain of our algorithm over two baseline algorithms (which ignore the temporal correlation and correlation between nodes), using synthetic experiments and real-world data. These two baseline algorithms representing the current approaches for processing event stream data. We also show that our algorithm is very sensitive to true changes, and the theoretical false-alarm-rates are very accurate compared to the experimental results.

3.1.1 Related work.

Recently, there has been a surge of interests in using multidimensional point processes for modeling dynamic event data over networks. However, most of these works focus on modeling and inference of the point processes over networks. Related works include modeling and learning bursty dynamics [44]; shaping social activity by incentivization [54]; learning information diffusion networks [43]; inferring causality [55]; learning mutually exciting processes for viral diffusion [56]; learning triggering kernels for multi-dimensional Hawkes processes [57]; in networks where each dimension is a Poisson process [58]; learning latent network structure for general counting processes [59]; tracking parameters of dynamic point process networks [60]; and estimating point process models for the co-evolution of network structure an information diffusion [61], just to name a few. These existing works provide a wealth of tools through which we can, to some extent, keep track of the network dynamics if the model parameters can be sequentially updated. However, only given the values of the up-to-date model parameters, especially in high dimensional networks, it is still not clear how to perform change detection based on these models in a principled fashion.

Classical statistical sequential analysis (see, e.g., [62, 63]), where one monitors *i.i.d.*

univariate and low-dimensional multivariate observations observations from a single data stream is a well-developed area. Outstanding contributions include Shewhart's control chart [64], the minimax approach Page's CUSUM procedure [65, 66], the Bayesian approach Shiryaev-Roberts procedure [67, 68], and window-limited procedures [69]. However, there is limited research in monitoring large-scale data streams over a network, or even event streams over networks. Detection of change-points in point processes has so far mostly focused on the simple Poisson process models without considering temporal dependency, and most of the detection statistics are computed in a discrete-time fashion, that is, one needs to aggregate the observed events into bins and then apply the traditional detection approaches to time-series of count data. Examples include [70, 71, 41] .

The notations are standard. The remaining sections are organized as follows. Section 3.2 presents the point process model and derives the likelihood functions. Section 3.3 presents our sequential likelihood ratio procedure. Section 3.4 presents the EM-like algorithm. Section 3.5 presents our theoretical approximation to false-alarm-rate. Section 3.6 contains the numerical examples. Section 3.6 presents our results for real-data. Finally, Section 3.8 summarizes the paper. All proofs are delegated to the Appendix.

3.2 Model and Formulation

Consider a sequence of events over a network with d nodes, represented as a double sequence

$$(t_1, u_1), (t_2, u_2), \ldots, (t_n, u_n), \ldots \tag{3.1}$$

where $t_i \in \mathbb{R}^+$ denotes the real-valued time when the ith event happens, and $i \in \mathbb{Z}^+$ and $u_i \in \{1, 2, \ldots, d\}$ indicating the node index where the event happens. We use temporal point processes [72] to model the discrete event streams, since they provide convenient tool in directly modeling the time intervals between events, and avoid the need of picking a time window to aggregate events and allow temporal events to be modeled in a fine grained fashion.

3.2.1 Temporal point processes

A temporal point process is a random process whose realization consists of a list of discrete events localized in time, $\{t_i\}$, with $t_i \in \mathbb{R}^+$ and $i \in \mathbb{Z}^+$. We start by considering one-dimensional point processes. Let the list of times of events up to but not including time t be the history

$$\mathcal{H}_t = \{t_1, \ldots, t_n : t_n < t\}.$$

Let N_t represent the total number of events till time t. Then the counting measure can be defined as

$$dN_t = \sum_{t_i \in \mathcal{H}_t} \delta(t - t_i) dt, \tag{3.2}$$

where $\delta(t)$ is the Dirac function.

To define the likelihood ratio for point processes, we first introduce the notion of *conditional intensity function* [73]. The conditional intensity function is a convenient and intuitive way of specifying how the present depends on the past in a temporal point process. Let $F^*(t)$ be the conditional probability that the next event t_{n+1} happens before t given the history of previous events

$$F^*(t) = \mathbb{P}\{t_{n+1} < t | \mathcal{H}_t\},$$

and let $f^*(t)$ be the corresponding conditional density function. The conditional intensity function (or the hazard function) [73] is defined by

$$\lambda_t = \frac{f^*(t)}{1 - F^*(t)}, \tag{3.3}$$

and it can be interpreted as the probability that an event occurs in an infinitesimal interval

$$\lambda_t dt = \mathbb{P}\{\text{event in } [t, t + dt) | \mathcal{H}_t\}. \tag{3.4}$$

This general model includes Poisson process and Hawkes process as special cases.

(*i*) For (inhomogeneous) Poisson processes, each event is stochastically independent to all the other events in the process, and the time intervals between consecutive events are independent with each other and are exponentially distributed. As a result, the conditional intensity function is independent of the past, which is simply deterministic $\lambda_t = \mu_t$.

(*ii*) For one dimensional Hawkes processes, the intensity function is history dependent and models a mutual excitation between events

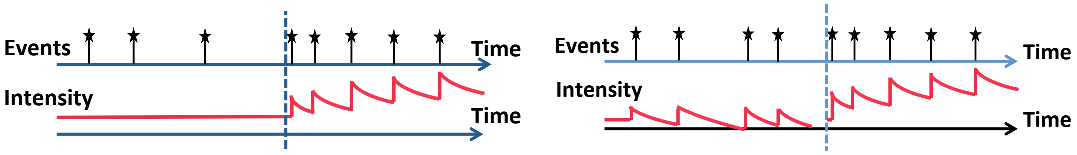
$$\lambda_t = \mu_t + \alpha \int_0^t \varphi(t-\tau) dN_\tau, \tag{3.5}$$

where μ_t is the base intensity (deterministic), $\alpha \in (0,1)$ (due to the requirement of stationary condition) is the influence parameter, and $\varphi(t)$ is a normalized kernel function $\int \varphi(t)dt = 1$. Together, they characterize how the history influences the current intensity. Fixing the kernel function, a higher value of α means a stronger temporal dependency between events. A commonly used kernel function is the exponential kernel $\varphi(t) = \beta e^{-\beta t}$, which we will use through the paper.

(*iii*) The multi-dimensional Hawkes process is defined similarly, with each dimension being a one-dimensional counting process. It can be used to model the sequence of events over network such as (3.1). We may convert a multi-dimensional process into a double sequence, using the first coordinate to represent time of the event, and the second coordinate to represent the index of the corresponding node.

Define a multivariate counting process $(N_t^1, N_t^2, \ldots, N_t^d)$, $t \geqslant 0$, with each component N_t^i recording the number of events of the i-th component (node) of the network during $[0,t]$. The intensity function is

$$\lambda_t^i = \mu_t^i + \sum_{j=1}^{d} \int_0^t \alpha_{ij} \varphi(t-\tau) dN_\tau^j, \tag{3.6}$$



(a) Poisson to Hawkes

(b) Hawkes to Hawkes

Figure 3.2: Illustration of scenarios for one-dimensional examples: (a) Poisson to Hawkes; (b) Hawkes to Hawkes.

where α_{ij}, $j, i \in \{1, \ldots, d\}$ represents the strength of influence of the j-th node on the i-th node by affecting its intensity process λ^i. If $\alpha_{ij} = 0$, then it means that N^j is not influencing N^i. Written in matrix form, the intensity can be expressed as

$$\boldsymbol{\lambda}_t = \boldsymbol{\mu}_t + \boldsymbol{A} \int_0^t \varphi(t-\tau) d\boldsymbol{N}_\tau, \tag{3.7}$$

where

$$\boldsymbol{\mu}_t = [\mu_t^1, \mu_t^2, \ldots, \mu_t^d]^\top, d\boldsymbol{N}_\tau = [dN_\tau^1, dN_\tau^2, \ldots, dN_\tau^d]^\top,$$

and $\boldsymbol{A} = [\alpha_{ij}]_{1 \leqslant i,j \leqslant d}$ is the *influence matrix*, which is our main quantity-of-interest when detect a change. The diagonal entries characterize the self-excitation and the off-diagonal entries capture the mutual-excitation among nodes in the network. The influence matrix can be asymmetric since influence can be bidirectional.

3.2.2 Likelihood function

In the following, we will explicitly denote the dependence of the likelihood function on the parameters in each setting. The following three cases are useful for our subsequent derivations. Let $f(t)$ denote the probability density function. For the one-dimensional setting, given a sequence of n events (event times) $\{t_1, t_2, \ldots, t_n\}$ before time t. Using the conditional probability formula, we obtain

$$\begin{aligned}\mathcal{L} = f(t_1, \dots, t_n) &= (1 - F^*(t)) \prod_{i=1}^{n} f(t_i | t_1, \dots, t_{i-1}) \\ &= (1 - F^*(t)) \prod_{i=1}^{n} f^*(t_i) = \left(\prod_{i=1}^{n} \lambda_{t_i} \right) \text{exp} \left\{ - \int_0^t \lambda_s ds \right\}.\end{aligned} \tag{3.8}$$

The last equation is from the following argument. From the definition of the conditional density function, we have

$$\lambda_t = \frac{d}{dt} F^*(t) / (1 - F^*(t)) = -\frac{d}{dt} \text{log}(1 - F^*(t)).$$

Hence, $\int_{t_n}^{t} \lambda_s ds = -\text{log}(1 - F^*(t))$, where $F^*(t_n) = 0$, since event $n + 1$ cannot happen at time t_n. Therefore,

$$F^*(t) = 1 - \text{exp} \left\{ - \int_{t_n}^{t} \lambda_s ds \right\},\ f^*(t) = \lambda_t \text{exp} \left\{ - \int_{t_n}^{t} \lambda_s ds \right\}.$$

The likelihood function for multi-dimensional Hawkes process can be derived similarly, by redefining $f^*(t)$ and $F^*(t)$ according to the intensity functions of the multi-dimensional processes.

Based on the above principle, we can derive the following likelihood functions.

Homogeneous Poisson process

For homogeneous Poisson process, $\lambda_t = \mu$. Given constant intensity, the log-likelihood function for a list of events $\{t_1, t_2, \dots, t_n\}$ in the time interval $[0, t]$ can be written as

$$\log \mathcal{L}(\mu) = n \text{log}\, \mu - \mu t. \tag{3.9}$$

One dimensional Hawkes process

For one-dimensional Hawkes process with constant baseline intensity $\mu_t = \mu$ and exponential kernel, we may obtain its log-likelihood function based on the above calculation. By substituting the conditional intensity function (3.5) into (3.8), the log-likelihood function for events in the time interval $[0, t]$ is given by

$$\begin{aligned}\log \mathcal{L}(\alpha, \beta, \mu) = &\sum_{i=1}^{n} \log \left(\mu + \alpha \sum_{t_j < t_i} \beta e^{-\beta(t_i - t_j)} \right) \\ &- \mu t - \sum_{t_i < t} \alpha \left[1 - e^{-\beta(t - t_i)} \right].\end{aligned} \tag{3.10}$$

To obtain the above expression, we have used the following two simple results for exponential kernels, due to the property of counting measure defined in (3.2):

$$\lambda_t = \mu + \alpha \int_{-\infty}^{t} \varphi(t - \tau) dN_\tau = \mu + \alpha \sum_{t_i < t} \beta e^{-\beta(t - t_i)}, \tag{3.11}$$

and

$$\int_0^t \lambda_s ds = \mu t + \sum_{t_i < t} \alpha \left[1 - e^{-\beta(t - t_i)} \right]. \tag{3.12}$$

Multi-dimensional Hawkes process

For multi-dimensional point process, we consider the event stream such as (3.1). Assume base intensities are constants with $\mu_t^i \triangleq \mu_i$. Using similar calculations as above, we obtain the log-likelihood function for events in the time interval $[0, t]$ as

$$\begin{aligned}\log \mathcal{L}\left(\boldsymbol{A}, \beta, \boldsymbol{\mu}\right) = &\sum_{i=1}^{n} \log \left[\mu_{u_i} + \sum_{t_j < t_i} \alpha_{u_i, u_j} \beta e^{-\beta(t_i - t_j)} \right] \\ - &\sum_{j=1}^{d} \mu_j t - \sum_{j=1}^{d} \sum_{t_i < t} \alpha_{u_i, j} \left[1 - e^{-\beta(t - t_i)} \right].\end{aligned} \tag{3.13}$$

3.3 Sequential change-point detection

We are interested in detecting two types of changes sequentially from event streams, which capture two general scenarios in real applications (Fig. 3.2 illustrates these two scenarios for the one dimensional setting): (*i*) The sequence before change is a Poisson process and after the change is a Hawkes process. This can be useful for applications where we are interested in detecting an emergence of self- or mutual-excitation between nodes. (*ii*) The sequence before change is a Hawkes process and after the change the magnitude of influence matrix increases. This can be a more realistic scenario, since often nodes in a network will influence each initially. This can be useful for applications where a triggering event changes the behavior or structure of the network. For instance, detecting emergence of a community in network [74].

In the following, we cast the change-point detection problems as sequential hypothesis test [75], and derive generalized likelihood ratio (GLR) statistic for each case. Suppose there may exist an unknown change-point κ such that after that time, the distribution of the point process changes.

3.3.1 Change from Poisson to Hawkes

First, we are interested in detecting the events over network changing from d-dimensional independent Poisson processes to an intertwined multivariate Hawkes process. This models the effect that the change affects the spatial dependency structure over the network. Below, we first consider one-dimensional setting, and then generalize them to multi-dimensional case.

One-dimensional case

The data consists of a sequence of events occurring at time $\{t_1, t_2, \ldots, t_n\}$. Under the hypothesis of no change (i.e. H_0), the event time is a one-dimensional Poisson process with intensity λ. Under the alternative hypothesis (i.e. H_1), there exists a change-point κ. The sequence is a Poisson process with intensity λ initially, and changes to a one-dimensional Hawkes process with parameter α after the change. Formally, the hypothesis test can be stated as

$$\begin{cases} \mathsf{H}_0: & \lambda_s = \mu, \quad 0 < s < t; \\ \mathsf{H}_1: & \lambda_s = \mu, \quad 0 < s < \kappa, \\ & \lambda_s^* = \mu + \alpha \int_\kappa^s \varphi(s-\tau) dN_\tau, \quad \kappa < s < t. \end{cases} \tag{3.14}$$

Assume intensity μ can be estimated from reference data and β is given as a priori. We treat the post-change influence parameter α as unknown parameter since it represents an anomaly.

Using the likelihood functions derived in Section 3.2.2, equations (3.9) and (3.10), for a hypothetical change-point location τ, the log-likelihood ratio as a function of α, β and μ, is given by

$$\begin{aligned} \ell_{t,\tau,\alpha} &= \log \mathcal{L}(\alpha,\beta,\mu) - \log \mathcal{L}(\mu) \\ &= \sum_{t_i \in (\tau,t)} \log \left[\mu + \alpha \sum_{t_j \in (\tau, t_i)} \beta e^{-\beta(t_i - t_j)} \right] \\ &\quad - \mu(t-\tau) - \alpha \sum_{\tau_i \in (\tau,t)} \left[1 - e^{-\beta(t-t_i)} \right]. \end{aligned} \tag{3.15}$$

Note that log-likelihood ratio only depends on the events in the interval (τ, t) and α. We maximize the statistic with respect to the unknown parameters α and τ to obtain the log GLR statistic. Finally, the sequential change-point detection procedure is a stopping rule (related to the non-Bayesian minimax type of detection rule, see [76]):

$$T_{\text{one-dim}} = \inf\{t : \max_{\tau < t} \max_{\alpha} \ \ell_{t,\tau,\alpha} > x\}, \tag{3.16}$$

where x is a pre-scribed threshold, whose choice will be discussed later. Even though there does not exist a closed-form expression for the estimator of α, we can estimate α via an

EM-like algorithm, which will be discussed in Section 3.4.2.

Remark 6 (Offline detection) *We can adapt the procedure for offline change-point detection by considering the fixed-sample hypothesis test. For instance, for the one-dimensional setting, given a sequence of n events with $t_{\max} \triangleq t_n$, we may detect the existence of change when the detection statistic, $\max_{\tau < t_{\max}} \max_\alpha \ \ell_{t_{\max},\tau,\alpha}$, exceeds a threshold. The change-point location can be estimated as τ^* that obtains the maximum. However, the algorithm consideration for online and offline detection are very different, as discussed in Section 3.4.*

Multi-dimensional case

For the multi-dimensional case, the event stream data can be represented as a double sequence defined in (3.1). We may construct a similar hypothesis test as above. Under the hypothesis of no change, the event times is multi-dimensional Poisson process with a vector intensity function $\boldsymbol{\lambda}_s = \boldsymbol{\mu}$. Under the alternative hypothesis, there exists a change-point κ. The sequence is a multi-dimensional Poisson process initially, and changes to a multi-dimensional Hawkes process with influence matrix $\boldsymbol{A}$ afterwards. We omit the formal statement of the hypothesis test as it is similar to (3.14).

Again, using the likelihood functions derived in 3.2.2, we obtain the likelihood ratio. The log-likelihood ratio for data up to time t, given a hypothetical change-point location τ and parameter $\boldsymbol{A}$, is given by

$$\begin{aligned}\ell_{t,\tau,\boldsymbol{A}} &= \log \mathcal{L}(\boldsymbol{A}, \beta, \mu) - \log \mathcal{L}(\mu) \\ &= \sum_{t_i \in (\tau, t)} \log \left[1 + \frac{1}{\mu_{u_i}} \sum_{t_j \in (\tau, t_i)} \alpha_{u_i,u_j} \beta e^{-\beta(t_i - t_j)} \right] \\ &\quad - \sum_{j=1}^{d} \sum_{t_i \in (\tau, t)} \alpha_{j,u_i} \left[1 - e^{-\beta(t - t_i)} \right].\end{aligned} \tag{3.17}$$

The sequential change-point detection procedure is a stopping rule:

$$T_{\text{multi-dim}} = \inf\{t : \max_{\tau<t} \max_{\boldsymbol{A}} \ \ell_{t,\tau,\boldsymbol{A}} > x\}, \tag{3.18}$$

where x is a pre-determined threshold. The multi-dimensional maximization can be computed efficiently via an EM algorithm described in Section 3.4.2 .

Remark 7 (Topology of network) *The topology of the network has been embedded in the sparsity pattern of the influence matrix A, which are given as a priori. The dependency between different nodes in the network and the temporal dependence over events can be captured in updating (or tracking) the influence matrix A with events stream. This can be achieved as an EM-like algorithm, which is resulted from solving a sequential optimization problem with warm start (i.e., we always initialize the parameters using the optimal solutions of the last step).*

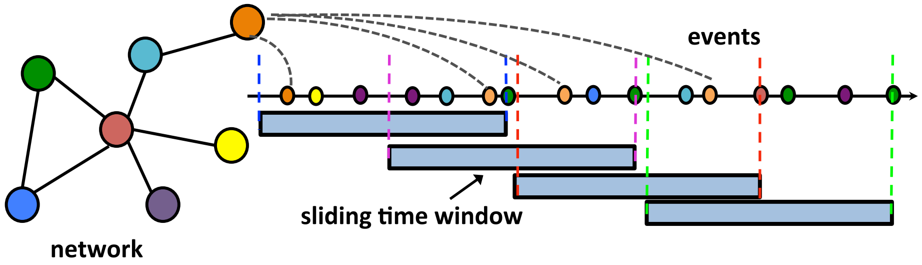


Figure 3.3: Illustration of the sliding window approach for online detection.

3.3.2 Changes from Hawkes to Hawkes

Next, consider the scenario where the process prior to change is a Hawkes process, and the change happens in the influence parameter α or the influence matrix $\boldsymbol{A}$.

One-dimensional case

Under the hypothesis of no change, the event stream is a one-dimensional Hawkes process with parameter α. Under the alternative hypothesis, there exists a change-point κ. The sequence is a Hawkes process with intensity α, and after the change, the intensity changes to α^*. Assume the parameter α prior to change is known.

Using the likelihood functions derived in 3.2.2, we obtain the log-likelihood ratio

$$
\begin{aligned}
\ell_{t,\tau,\alpha^*} &= \log \mathcal{L}(\alpha^*, \beta, \mu) - \log \mathcal{L}(\mu) \\
&= \sum_{t_i \in (\tau,t)} \log \left[\frac{\mu + \alpha^* \sum_{t_j \in (\tau, t_i)} \beta e^{-\beta(t_i - t_j)}}{\mu + \alpha \sum_{t_j \in (\tau, t_i)} \beta e^{-\beta(t_i - t_j)}} \right] \\
&\quad - (\alpha^* - \alpha) \sum_{t_i \in (\tau,t)} \left[1 - e^{-\beta(t - t_i)} \right],
\end{aligned} \tag{3.19}
$$

and the change-point detection is through a procedure in the form of (3.16) by maximizing with respect to τ and α.

Multi-dimensional case

For the multi-dimensional setting, we assume the change will alter the influence parameters of the multi-dimensional Hawkes process over network. This captures the effect that, after the change, the influence between nodes becomes different. Assume that under the hypothesis of no change, the event stream is a multi-dimensional Hawkes process with parameter $\boldsymbol{A}$. Alternatively, there exists a change-point κ. The sequence is a multi-dimensional Hawkes process with influence matrix $\boldsymbol{A}$ before the change, and after the change, the influence matrix becomes $\boldsymbol{A}^*$. Assume the influence matrix $\boldsymbol{A}$ prior to change is known.

Using the likelihood functions derived in 3.2.2, the log-likelihood ratio at time t for a hypothetical change-point location τ and post-change parameter value $\boldsymbol{A}^*$ is given by

$$
\begin{aligned}
\ell_{t,\tau,\boldsymbol{A}^*} &= \log \mathcal{L}(\boldsymbol{A}^*, \beta, \mu) - \log \mathcal{L}(\mu) \\
&= \sum_{t_i \in (\tau,t)} \log \left[\frac{\mu_{u_i} + \sum_{t_j \in (\tau, t_i)} \alpha^*_{u_i,u_j} \beta e^{-\beta(t_i - t_j)}}{\mu_{u_i} + \sum_{t_j \in (\tau, t_i)} \alpha_{u_i,u_j} \beta e^{-\beta(t_i - t_j)}} \right] \\
&\quad - \sum_{j=1}^{d} \sum_{t_i \in (\tau,t)} \left(\alpha^*_{j,u_i} - \alpha_{j,u_i} \right) \left[1 - e^{-\beta(t - t_i)} \right],
\end{aligned} \tag{3.20}
$$

and the change-point detection is through a procedure in the form of (3.18) by maximizing with respect to τ and $\boldsymbol{A}^*$.

Algorithm 1 Online Detection Algorithm

```
Require: Data $\{(t_i, u_i)\}$. Scanning window length $L$; Update frequency $\gamma$ (per events);
    Initialization for parameters $\alpha$ (one-dimension) or $\boldsymbol{A}$ (multi-dimension); Pre-defined
    threshold: $x$; Estimation accuracy: $\epsilon$.
repeat
    if $\mod(i, \gamma) = 0$ then
        Initialize $\alpha^{(0)} = \hat{\alpha}$ or $\boldsymbol{A}^{(0)} = \hat{\boldsymbol{A}}$ {warm start}
        repeat
            Perform {E-step} and {M-step} from Section 3.4.2
        until $\|\alpha^{(k+1)} - \alpha^{(k)}\| < \epsilon$ or $\|\boldsymbol{A}^{(k+1)} - \boldsymbol{A}^{(k)}\| < \epsilon$
        Let $\hat{\alpha} = \alpha^{(k+1)}$ and $\hat{\boldsymbol{A}} = \boldsymbol{A}^{(k+1)}$.
        Use $\hat{\alpha}$ or $\hat{\boldsymbol{A}}$ to compute log likelihood using (3.15), (3.17), (3.19) or (3.20).
    end if
until $\ell_{t,\tau,\hat{\alpha}} > x$ or $\ell_{t,\tau,\hat{\boldsymbol{A}}} > x$ and announce a change.
```

3.4 Algorithm for computing likelihood online

In the online setting, we obtain new data continuously. Hence, in order to perform online detection, we need to update the likelihood efficiently to incorporate the new data. To reduce computational cost, update of the likelihood function can be computed recursively and the update algorithm should have low cost. To reduce memory requirement, the algorithm should only store the minimum amount of data necessary for detection rather than the complete history. These requirements make online detection drastically different from offline detection. Since in the offline setting, we can afford more computational complexity.

3.4.1 Sliding window procedure

The basic idea of online detection procedure is illustrated in Fig. 3.3. We adopt a *sliding window* approach to reduce computational complexity as well the memory requirement. When evaluating the likelihood function, instead of maximizing over possible change-point location $\tau < t$, we pick a window-size L and set τ to be a fixed-value

$$\tau = t - L.$$

This is equivalent to constantly testing whether a change-point occurs L samples before. By fixing the window-size, we reduce the computational complexity, since we eliminate the maximization over the change-point location. This also reduces the memory requirement as we only need to store events that fall into the sliding window. The drawback is that, by doing this, some statistical detection power is lost, since we do not use the most likely change-point location, and it may increase detection delay. When implementing the algorithm, we choose L to achieve a good balance in these two aspect. We have to choose L large enough so that there is enough events stored for us to make a consistent inference. In practice, a proper length of window relies on the nature of the data. If the data are noisy, usually a longer time window is needed to have a better estimation of the parameter and reduce the false alarm.

3.4.2 Parameter Free EM-like Algorithm

We consider one-dimensional point process to illustrate the derivation of the EM-like algorithm. It can be shown that the likelihood function (3.15) is a concave function with respect to the parameter α. One can use gradient descent to optimize this objective, where the algorithm will typically involves some additional tuning parameters such as the learning rate. Although there does not exist a closed-form estimator for influence parameter α or influence matrix $\boldsymbol{A}$, we develop an efficient EM algorithm to update the likelihood, exploiting the structure of the likelihood function [77]. The overall algorithm is summarized in Algorithm 1.

First, we obtain a concave lower bound of the likelihood function using Jensen's inequality. Consider all events fall into a sliding window $t_i \in (\tau, t)$ at time t. Introduce auxiliary variables p_{ij} for all pair of events (i, j) within the window and such that $t_j < t_i$. The variables are subject to the constraint

$$\forall i, \sum_{t_j<t_i} p_{ij} = 1, \quad p_{ij} \geqslant 0. \tag{3.21}$$

These p_{ij} can be interpreted as the probability that j-th event influence the i-th event in the sequence. It can be shown that the likelihood function defined in (3.10) can be lower-bounded

$$\begin{aligned} \ell_{t,\tau,\alpha} \geqslant \sum_{t_i\in(\tau,t)} &\left(p_{ii}\text{log}(\mu) + \sum_{t_j\in(\tau,t_i)} p_{ij}\text{log}\left[\alpha\beta e^{-\beta(t_i-t_j)}\right] \right. \\ &\left. - \sum_{t_j\in(\tau,t)} p_{ij}\text{log}p_{ij} \right) - \mu(t-\tau) - \alpha \sum_{t_i\in(\tau,t)} \left[1 - e^{-\beta(t-t_i)}\right], \end{aligned}$$

Note that the lower-bound is valid for every choice of $\{p_{ij}\}$ which satisfies (3.21).

To make the lower bound tight and ensure improvement in each iteration, we will maximize it with respect to p_{ij} and obtain (3.22) (assuming we have $\alpha^{(k)}$ from previous iteration or initialization). Once we have the tight lower bound, we will take gradient of this lower-bound with respect to α. When updating from the k-th iteration to the $(k + 1)$-th iteration, we obtain (3.23)

$$p_{ij}^{(k)} = \frac{\alpha^{(k)}\beta e^{-\beta(t_j-t_i)}}{\mu + \alpha^{(k)}\beta \sum_{t_m\in(\tau,t_j)} e^{-\beta(t_j-t_m)}} \quad \{\text{E-step}\} \tag{3.22}$$

$$\alpha^{(k+1)} = \frac{\sum_{i<j} p_{ij}^{(k)}}{\sum_{t_i\in(\tau,t)}[1 - e^{-\beta(t-t_i)}]} \quad \{\text{M-step}\} \tag{3.23}$$

where the superscript denotes the number of iterations. The algorithm iterates these two steps until the algorithm converges and obtains the estimated α. In practice, we find that we only need 3 or 4 iterations to converge if using warm start.

Similarly, online estimate for the influence matrix for multi-dimensional case can be estimated by iterating the following two steps:

$$p_{ij}^{(k)} = \frac{\alpha_{u_i,u_j}^{(k)} \beta e^{-\beta(t_i - t_j)}}{\mu_{u_i} + \beta \sum_{t_m \in (\tau, t_i)} \alpha_{u_i,u_m}^{(k)} e^{-\beta(t_i - t_m)}} \quad \{\text{E-step}\}$$

$$\alpha_{u,v}^{(k+1)} = \frac{\sum_{i:\, u_i = u} \sum_{j<i:\, u_j = v} p_{ij}^{(k)}}{\sum_{j:\, t_j \in (\tau, t), u_j = v} \left[1 - e^{-\beta(t - t_j)}\right]}. \quad \{\text{M-step}\}$$

The overall detection procedure is summarized in Fig. 3.3 and Algorithm 1.

Remark 8 (Computational complexity) *The key computation is to compute pairwise inter-event times for pairs of event $t_i - t_j$, $i < j$. It is related to the window size (since we have adopted a sliding window approach), the size of the network, and the number of EM steps. However, note that in the EM algorithm, we only need to compute the inter-event times for nodes that are connected by an edge, since the summation is weighted by α_{ij} and the term only counts if α_{ij} is non-zero. Hence, the updates only involve neighboring nodes and the complexity is proportional to the number of edges in the network. Since most social networks are sparse, the will significantly lower the complexity. We may reduce the number of EM iterations for each update, by leveraging a warm-start for initializing the parameter values: since typically for two adjacent sliding window, the corresponding optimal parameter values should be very close to the previous one.*

Remark 9 (Distributed implementation) *Our EM-like algorithm in the network setting can be implemented in a distributed fashion. This has embedded in the form of the algorithm already. Hence, the algorithm can be used for process large networks. In the E-step, when updating the p_{ij}, we need to evaluate a sum in the denominator, and this is the only place where different nodes need to exchange information, i.e., the event times happened at that node. Since we only need to sum over all events such that the corresponding α_{u_i,u_j} is non-zero, this means that each node only needs to consider the events happened at the neighboring nodes. Similarly, in the M-step, only neighboring nodes need to exchange their values of p_{ij} and event times to update the influence parameter values.*

3.5 Theoretical threshold

A key step in implementing the detection algorithm is to set the threshold. The choice of threshold involves a trade-off between two standard performance metrics for sequential change-point detection: the false-alarm rate and how fast we can detect the change. Formally, these two performance metrics are: (i) the expected stopping time when there is no change-points, or named average run length (ARL); and (ii) the expected detection delay when there exists a change-point.

Typically, a higher threshold x results in a larger ARL (hence smaller false-alarm rate) but larger detection delay. A usual practice is to set the false-alarm-rate (or ARL) to a pre-determined value, and find the corresponding threshold x. The pre-determined ARL depends on how frequent we can tolerate false detection (once a month or once a year). Usually, the threshold is estimated via direct Monte Carlo by relating threshold to ARL assuming the data follow the null distribution. However, Monte Carlo is not only computationally expensive, in some practical problems, repeated experiments would be prohibitive. Therefore it is important to find a cheaper way to accurately estimate the threshold.

We develop an analytical function which relates the threshold to ARL. That is, given a prescribed ARL, we can solve for the corresponding threshold x analytically. We first characterize the property of the likelihood ratio statistic in the following lemma, which states that the mean and variance of the log-likelihood ratios both scale roughly linearly with the post-change time duration. This property of the likelihood ratio statistics is key to developing our main result.

Lemma 10 (Mean and variance of log-likelihood ratios) *When the number of post-change samples $(t-\tau)$ is large, the mean and variance of log-likelihood ratio for the single-dimensional and the multi-dimensional cases, denoted as $\ell_{t,\tau,\cdot}$, for our cases converges to simple linear form. Under the null hypothesis, $\mathbb{E}[\ell_{t,\tau,\cdot}] \approx (t-\tau)I_0$ and $\mathbb{E}[\ell_{t,\tau,\cdot}] \approx (t-\tau)\sigma_0^2$. Under the alternative hypothesis, $\mathbb{E}[\ell_{t,\tau,\cdot}] \approx (t-\tau)I$ and $\mathbb{E}[\ell_{t,\tau,\cdot}] \approx (t-\tau)\sigma^2$. Above, I,*

I_0, σ^2, and σ_0^2 are defined in Table 3.1 for various settings we considered.

Our main theoretical result is the following general theorem that can be applied for all hypothesis test we consider. Denote the probability and the expectation under the hypothesis of no change by $\mathbb{P}^\infty$ and $\mathbb{E}^\infty$, respectively.

Theorem 11 (ARL under the null distribution) *When $x \to \infty$ and $x/\sqrt{L} \to c'$ for some constant c', the average run length (ARL) of the stopping time T defined in (3.16) for one-dimensional case, is given by*

$$\mathbb{E}^\infty[T_{\text{one-dim}}] = e^x \left[\int_\alpha \nu\left(\frac{2\xi}{\eta^2}\right) \frac{\phi\left(\frac{LI-x}{\sqrt{L\sigma^2}}\right)}{\sqrt{L\sigma^2}} d\alpha \right]^{-1} \cdot (1 + o(1)). \tag{3.24}$$

For multi-dimensional case, the same expression holds for $\mathbb{E}^\infty[T_{\text{multi-dim}}]$ except that $\int_\alpha$ is replaced by $\int_{\boldsymbol{A}}$, which means taking integral with respect to all nonzero entries of the matrix $\int_{\boldsymbol{A}} = \int \cdots \int \int_{\{\alpha_{ij}, \alpha_{ij} \neq 0\}}$. Above, the special function

$$\nu(\mu) \approx \frac{(2/\mu)\,(\Phi(\mu/2) - 0.5)}{(\mu/2)\Phi(\mu/2) + \phi(\mu/2)}.$$

The specific expressions for I, I_0, σ^2, and σ_0^2 for various settings are summarized in Table 3.1, and

$$\xi = -(I_0 - I), \quad \eta^2 = \sigma_0^2 + \sigma^2. \tag{3.25}$$

Above, $\Phi(x)$ and $\phi(x)$ are the cumulative distribution function (CDF) and the probability density function (PDF) of the standard normal, respectively.

Remark 12 (Evaluating integral) *The multi-dimensional integral can be evaluated using Monte Carlo method [78]. We use this approach for our numerical examples as well.*

Remark 13 (Interpretation) *The parameters I_0, I, σ_0^2 and σ^2 have the following interpre-*

tation

$$I_0 = \mathbb{E}[\ell_{t,\tau,\alpha}]/L, \quad \sigma_0^2 = \mathit{Var}[\ell_{t,\tau,\alpha}]/L,$$
$$I = \mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]/L, \quad \sigma^2 = \mathit{Var}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]/L, \tag{3.26}$$

which are the mean and the variance of the log-likelihood ratio under the null and the alternative distributions, per unit time, respectively. Moreover, I can be interpreted roughly as the Kullback-Leibler information per time for each of the hypothesis test we consider.

The proof of the Theorem 11 combines the recently developed change-of-measure techniques for sequential analysis, with properties the likelihood ratios for the point processes, mean field approximation for point processes, and Delta method [79].

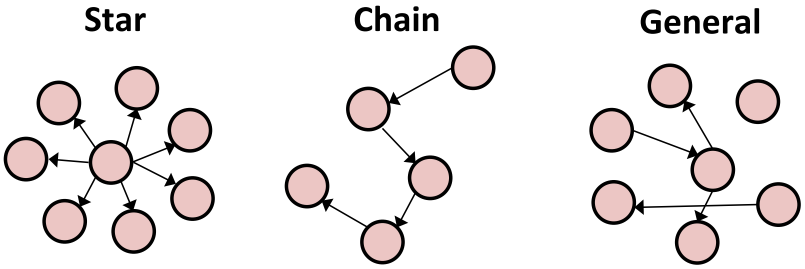


Figure 3.4: Illustration of network topology.

3.6 Numerical examples

In this section, we present some numerical experiments using synthetic data. We focus on comparing EDD of our algorithm with two baseline methods, and demonstrate the accuracy of the analytic threshold.

3.6.1 Comparison of EDD

Two baseline algorithms

We compare our method to two baseline algorithms:

Table 3.1: Expressions for I, I_0, σ^2 and σ_0^2 under different settings.

Setting	I	I_0	σ^2	σ_0^2
Poi. → Haw. (one dim.)	$\frac{\mu}{1-\alpha}\log\left(\frac{1}{1-\alpha}\right) - \frac{\alpha}{1-\alpha}\mu$	$\mu\log\left(\frac{1}{1-\alpha}\right) - \frac{\alpha}{1-\alpha}\mu$	$\left[\log\left(\frac{1}{1-\alpha}\right)\right]^2 \cdot$ $\left[\frac{\mu}{1-\alpha} + \frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3}\right]$	$\mu\left[\log\left(\frac{1}{1-\alpha}\right)\right]^2$
Poi. → Haw. (high dim.)	$\bar{\boldsymbol{\lambda}}^{*\intercal}\left(\log(\bar{\boldsymbol{\lambda}}^*) - \log(\boldsymbol{\mu})\right)$ $-\boldsymbol{e}^\intercal(\bar{\boldsymbol{\lambda}}^* - \boldsymbol{\mu})$	$\boldsymbol{\mu}^\intercal\left(\log(\bar{\boldsymbol{\lambda}}^*) - \log(\boldsymbol{\mu})\right)$ $-\boldsymbol{e}^\top(\bar{\boldsymbol{\lambda}}^* - \boldsymbol{\mu})$	$\boldsymbol{e}^\intercal\left(\boldsymbol{H} \circ \boldsymbol{C}\right)\boldsymbol{e}$	$\boldsymbol{\mu}^\intercal\left(\log(\bar{\boldsymbol{\lambda}}^*) - \log(\boldsymbol{\mu})\right)^{(2)}$
Haw. → Haw. (one dim.)	$\frac{\mu}{1-\alpha^*}\log\left(\frac{1-\alpha}{1-\alpha^*}\right)$ $-\frac{\mu}{1-\alpha^*} + \frac{\mu}{1-\alpha}$	$\frac{\mu}{1-\alpha}\log\left(\frac{1-\alpha}{1-\alpha^*}\right)$ $-\frac{\mu}{1-\alpha^*} + \frac{\mu}{1-\alpha}$	$\left[\log\left(\frac{1-\alpha}{1-\alpha^*}\right)\right]^2 \cdot$ $\left[\frac{\mu}{1-\alpha^*} + \frac{\alpha^*(2-\alpha^*)\mu}{(1-\alpha^*)^3}\right]$ $+\left(1 - \frac{1-\alpha}{1-\alpha^*}\right)^2 \cdot$ $\left[\frac{\mu}{1-\alpha} + \frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3}\right]$	$\left[1 - \frac{1-\alpha^*}{1-\alpha}\right]^2 \cdot$ $\left[\frac{\mu}{1-\alpha^*} + \frac{\alpha^*(2-\alpha^*)\mu}{(1-\alpha^*)^3}\right]$ $+\left[\log\left(\frac{1-\alpha}{1-\alpha^*}\right)\right]^2 \cdot$ $\left[\frac{\mu}{1-\alpha} + \frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3}\right]$
Haw. → Haw. (multi dim.)	$\bar{\boldsymbol{\lambda}}^{*\intercal}\left[\log\bar{\boldsymbol{\lambda}}^* - \log\bar{\boldsymbol{\lambda}}\right]$ $-\boldsymbol{e}^\intercal[\bar{\boldsymbol{\lambda}}^* - \bar{\boldsymbol{\lambda}}]$	$\bar{\boldsymbol{\lambda}}^\intercal\left[\log\bar{\boldsymbol{\lambda}}^* - \log\bar{\boldsymbol{\lambda}}\right]$ $-\boldsymbol{e}^\intercal[\bar{\boldsymbol{\lambda}}^* - \bar{\boldsymbol{\lambda}}]$	$\boldsymbol{e}^\intercal\left(\boldsymbol{G} \circ \boldsymbol{C}^* + \boldsymbol{F} \circ \boldsymbol{C}\right)\boldsymbol{e}$	$\boldsymbol{e}^\intercal\left(\boldsymbol{R} \circ \boldsymbol{C}^* + \boldsymbol{G} \circ \boldsymbol{C}\right)\boldsymbol{e}$

In the table above, $\boldsymbol{M}^{(2)} = \boldsymbol{M} \circ \boldsymbol{M}$ denote the Hadamard product, and related quantities are defined as

$$\begin{aligned}
&\bar{\boldsymbol{\lambda}}^* = (\boldsymbol{I} - \boldsymbol{A}^*)^{-1}\boldsymbol{\mu}, \quad \bar{\boldsymbol{\lambda}} = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}, \\
&\boldsymbol{H} = \left[\log\left((\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}\right) - \log\left(\boldsymbol{\mu}\right)\right] \cdot \left[\log\left((\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}\right) - \log\left(\boldsymbol{\mu}\right)\right]^\intercal, \\
&\boldsymbol{C} = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{A}\left(2\boldsymbol{I} + (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{A}\right)\operatorname{diag}\left((\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}\right) + \operatorname{diag}\left((\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}\right), \\
&\boldsymbol{C}^* = (\boldsymbol{I} - \boldsymbol{A}^*)^{-1}\boldsymbol{A}^*\left(2\boldsymbol{I} + (\boldsymbol{I} - \boldsymbol{A}^*)^{-1}\boldsymbol{A}^*\right) \cdot \operatorname{diag}\left((\boldsymbol{I} - \boldsymbol{A}^*)^{-1}\boldsymbol{\mu}\right) + \operatorname{diag}\left((\boldsymbol{I} - \boldsymbol{A}^*)^{-1}\boldsymbol{\mu}\right), \\
&\boldsymbol{G}_{ij} = \left[\log\left(\bar{\lambda}_i^*/\bar{\lambda}_i\right)\right] \cdot \left[\log\left(\bar{\lambda}_j^*/\bar{\lambda}_j\right)\right], \quad \boldsymbol{F}_{ij} = \left(1 - \bar{\lambda}_i^*/\bar{\lambda}_i\right)\left(1 - \bar{\lambda}_j^*/\bar{\lambda}_j\right), \\
&\boldsymbol{R}_{ij} = \left(\bar{\lambda}_i/\bar{\lambda}_i^* - 1\right)\left(\bar{\lambda}_j/\bar{\lambda}_j^* - 1\right), \quad 1 \leqslant i \leqslant j \leqslant d.
\end{aligned}$$

(i) **Baseline 1** is related to the commonly used "data binning" approach for processing discrete event data such as [45]. This approach, however, ignores the temporal correlation and correlation between nodes. Here, we convert the event data into counts, by discretize time into uniform grid, and count the number of events happening in each interval. Such counting data can be modeled via Poisson distribution. We may derive a likelihood ratio statistic to detect a change. Suppose $n_1, n_2, \ldots, n_c$ are the sequence of counting numbers following Poisson distribution with intensity λ_i, $i = 1, 2, \ldots, c$ is the index of the discrete time step. Assume under the null hypothesis, the intensity function is $\lambda_i = \mu$. Alternatively, there may exist a change-point κ such that before the change, $\lambda_i = \mu$, and after the change,

$\lambda_i = \mu^*$. It can be shown that the log-likelihood ratio statistic as

$$\ell_{c,k,\mu^*} = -(c-k)(\mu^* - \mu) + \sum_{i=k+1}^{c} n_i \log \frac{\mu^*}{\mu}.$$

We detect a change whenever $\max_{k<c} \max_{\mu^*} \ell_{k,c,\mu^*} > x$ for a pre-determined threshold x. Assume every dimension follows an independent Poisson process, then the log-likelihood ratio for the multi-dimensional case is just a summation of the log-likelihood ratio for each dimension. Suppose the total dimension is d, then

$$\ell_{k,c,\boldsymbol{\mu}^*} = \sum_{j=1}^{d} \left[-(c-k)(\mu_j^* - \mu_j) + \sum_{i=k+1}^{c} n_i^j \log \frac{\mu_j^*}{\mu_j} \right].$$

We detect a change whenever $\max_{k<c} \max_{\boldsymbol{\mu}^*} \ell_{k,c,\boldsymbol{\mu}^*} > x$.

(ii) **Baseline 2** method calculates the one-dimensional change-point detection statistic at each node separately as (3.15) and (3.19), and then combine the statistics by summation into a *global statistic* to perform detection. This approach, however, ignores the correlation between nodes, and can also be viewed as a *centralized* approach for change-point detection and it is related to multi-chart change-point detection [76].

Set-up of synthetic experiments

We consider the following scenarios and compare the EDD of our method to two baseline methods. EDD is defined as the average time (delay) it takes before we can detect the change, and can be understood as the power of the test statistic in the sequential setting. The thresholds of all the three methods are calibrated so that the ARL under the null model is 10^4 unit time and the corresponding thresholds are obtained via direct Monte Carlo for a fair comparison. The sliding window is set to be $L = 10$ unit time. The exponential kernel $\varphi(t) = \beta e^{-\beta t}$ is used and $\beta = 1$. The scenarios we considered are described below. The illustrations of the *Case 1* and *Case 2* scenarios are displayed in Fig. 3.2. The network topology for *Case 3* to *Case 7* are demonstrated in Fig. 3.4.

Case 1. Consider a situation when the events first follow a one-dimensional Poisson process with intensity $\mu = 10$ and then shift to a Hawkes process with influence parameter $\alpha = 0.5$. This scenario describes the emergence of temporal dependency in the event data.

Case 2. The process shifts from a one-dimensional Hawkes process with parameter $\mu = 10$, $\alpha = 0.3$ to another Hawkes process with a larger influence parameter $\alpha = 0.5$. The scenario represents the change of the temporal dependency in the event data.

Case 3. Consider a star network scenario with one parent and nine children, which is commonly used in modeling how the information broadcasting over the network. Before the change-point, each note has a base intensity $\mu = 1$ and the self-excitation $\alpha_{i,i} = 0.3$, $1 \leq i \leq 10$. The mutual-excitation from the parent to each child is set to be $\alpha_{1,j} = 0.3$, $2 \leq j \leq 10$ (if we use the first node to represent the parent). After the change-point, all the self- and mutual- excitation increase to 0.5.

Case 4. The network topology is the same as Case 3. But we consider a more challenging scenario. Before the change, parameters are set to be the same as Case 3. After the change, the self-excitation $\alpha_{i,i}$, $1 \leq i \leq 10$ deteriorate to 0.01, and the influence from the parent to the children increase to $\alpha_{1,j} = 0.6$, $j = 2 \leq j \leq 10$. In this case, for each note, the occurring frequency of events would be almost the same before and after the change-points. But the influence structure embedded in the network has actually changed.

Case 5. Consider a network with a chain of ten nodes, which is commonly used to model information propagation over the network. Before the change, each note has a base intensity $\mu = 1$ and the self-excitation $\alpha_{i,i} = 0.3$, $1 \leq i \leq 10$ and mutual-excitation $\alpha_{i,j} = 0.3$, where $j - i = 1, 1 \leq i \leq 9$. After the change-point, all the self- and mutual-excitation parameters increase to 0.5.

Case 6. Consider a *sparse* network with an arbitrary topology and one hundred nodes. Each note has a base intensity $\mu = 0.1$ and the self-excitation $\alpha_{i,i} = 0.3$, $1 \leq i \leq 100$. We randomly select twenty directed edges over the network and set the mutual-excitation to be $\alpha_{i,j} = 0.3$, where $i \neq j, i, j$ are randomly selected. After the change-point, all the self- and mutual-excitation increase to 0.5.

Case 7. The *sparse* network topology and the pre-change parameters are the same with Case 6. The only difference is that after the change-point, only half of the self- and mutual-excitation parameters increase to 0.5.

EDD results and discussions

For the above scenarios, we compare the EDD of our method and two baseline algorithms. The results are shown in Table 3.2. We see our method compares favorably to the two baseline algorithms. In the first five cases, our method has a significant performance gain. Especially for Case 4, which is a challenging setting, only our method succeeds in detecting the spatial structure changes. For Case 6 and Case 7, our method has similar performance as Baseline 2. A possible reason is that in these cases the network topology is a sparse graph so the nodes are "loosely" correlated. Hence, the advantage of combining over graph is not significant in these cases.

Moreover, we observe that Baseline 1 algorithm is not stable. In certain cases (Case 6 and Case 7), it completely fails to detect the change. An explanation is that there is a chance that the number of events fall into a given time bin is extremely small or close to zero, and this causes numerical issues when calculating the the likelihood function (since there is a log function of the number of events). On the other hand, our proposed log-likelihood ratio is event-triggered, and hence will avoid such numerical issues.

Table 3.2: EDD comparison. Thresholds for all methods are calibrated such that $ARL = 10^4$.

	Baseline 1	**Baseline 2**	**Our Method**
Case 1	22.1	–	**4.8**
Case 2	19.6	–	**18.8**
Case 3	8.2	6.9	**4.3**
Case 4	×	×	**19.8**
Case 5	6.1	5.7	**4.7**
Case 6	×	**10.5**	10.8
Case 7	×	**32.5**	**32.5**

Note: '×' means the corresponding method fails to detect the changes; '–' means in one-dimensional case Baseline 2 is identical to ours.

3.6.2 Sensitivity analysis

We also perform the sensitivity analysis by comparing our method to Baseline 1 algorithm via numerical simulation. The comparison is conducted under various kernel decay parameter β, and the strength of the post-change signals, which can be controlled by the magnitudes of the changes in α (or $\boldsymbol{A}$). For each dataset, we created 500 samples of sequences with half of them containing one true change-point and half of them containing no change-point. We then plot the *area under the curve* (AUC) (defined as the true positive rate versus the false positive rate under various threshold) for comparison, as shown in Fig. 3.5.

Set-up of synthetic experiments

Overall, we consider various decay parameter β and the magnitudes of the changes in α to compare the approaches.

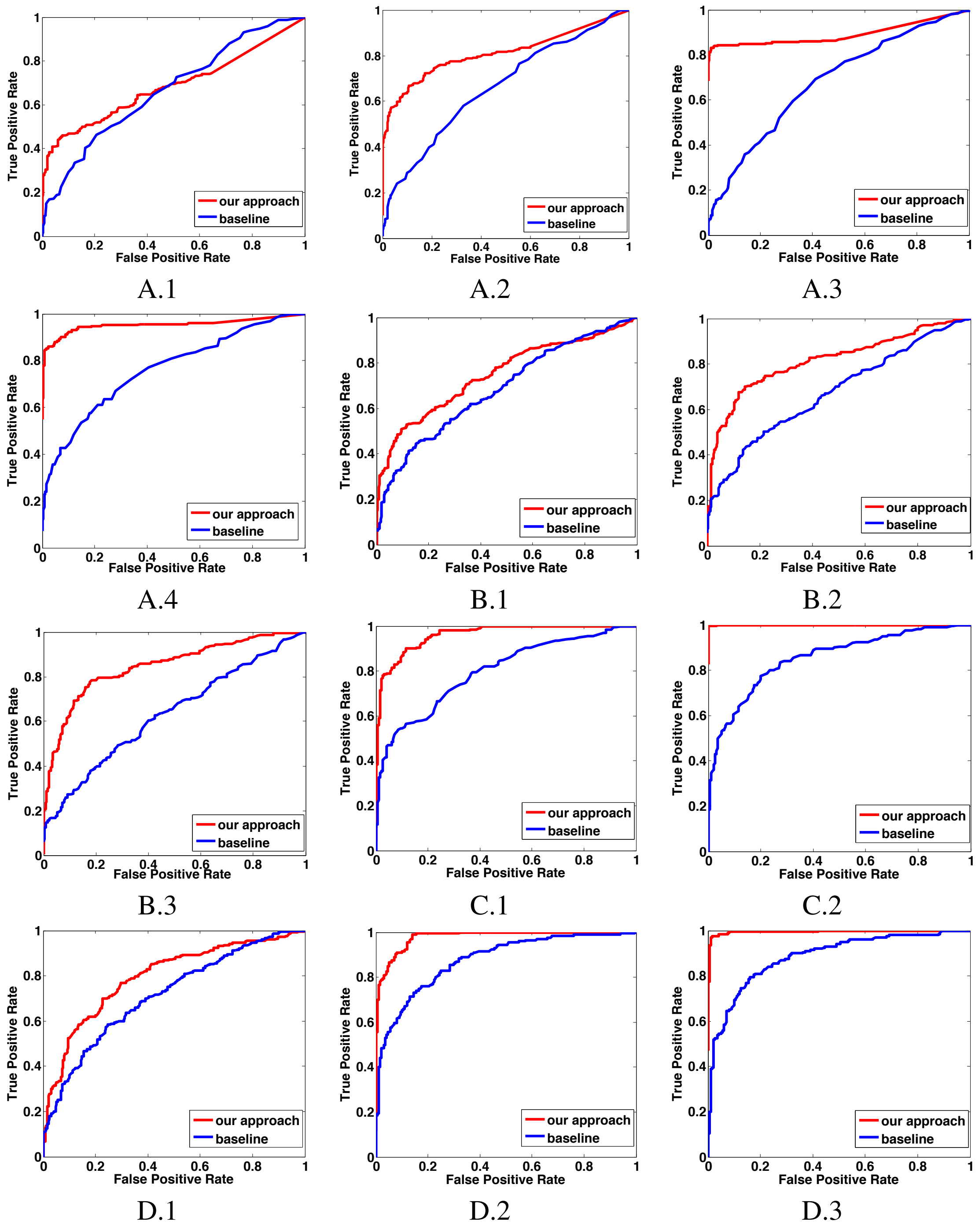
One-dimensional setting. First, consider that before the change the data is a Poisson process with base intensity $\mu = 1$. For A.1-A.4, the post-change data become one dimensional Hawkes process: for A.1–A.3, $\alpha = 0.2$, and $\beta = 1, 10, 100$, respectively; for A.4, $\alpha = 0.3$, and $\beta = 10$. By comparing the AUC curves, we see that, our method has a remarkably better performance in distinguishing the true positive changes from the false positive changes compared to the baseline method. The superiority would become more evident under larger β and bigger magnitudes of shifts in α. For weak changes, the baseline approach is just slightly better than the random guess, whereas our approach consistently performs well. Similar results can be found if the pre-change data follow the Hawkes process. For example, in B.1-B.3, the pre-change data follow Hawkes process with $\mu = 1$, $\alpha = 0.3$, and $\beta = 1$, and the post-change parameters shift to a Hawkes process with $\alpha = 0.5$, and $\beta = 1, 10, 100$, respectively. We can see the similar trend as before by varying β and α.

Network setting. We first consider the two-dimensional examples in the following and get the same results. For C.1-C.2, the pre-change data follow two dimensional Poisson processes with $\boldsymbol{\mu} = [0.2, 0.2]^\intercal$, and the post-change data follow two dimensional Hawkes processes with influence parameter $\boldsymbol{A} = [0.1, 0.1; 0.1, 0.1]$, with $\beta = 1, 10$, respectively. For D.1–D.3, consider the star network with one parent and nine children. Before the change-point, for each node the base intensity is $\mu = 0.1$, $\beta = 1$, and the influence from the parent to each child is $\alpha = 0.3$. After the change, α changes to 0.4 for D.1, and α changes to 0.5, $\beta = 1, 10$ respectively for D.2 and D.3.

3.6.3 Accuracy of theoretical threshold

We evaluate the accuracy of our approximation in Theorem 11 by comparing the threshold obtained via Theorem 11 with the true threshold obtained by direct Monte Carlo. We consider various scenarios and parameter settings. We demonstrate the results in Fig. 3.6 and list the parameters below.

For Fig. 3.6-(a)(b)(c), the null distribution is one-dimensional Poisson process with intensity $\mu = 1$. We choose $\beta = 1$ as a priori, and vary the length of the sliding time



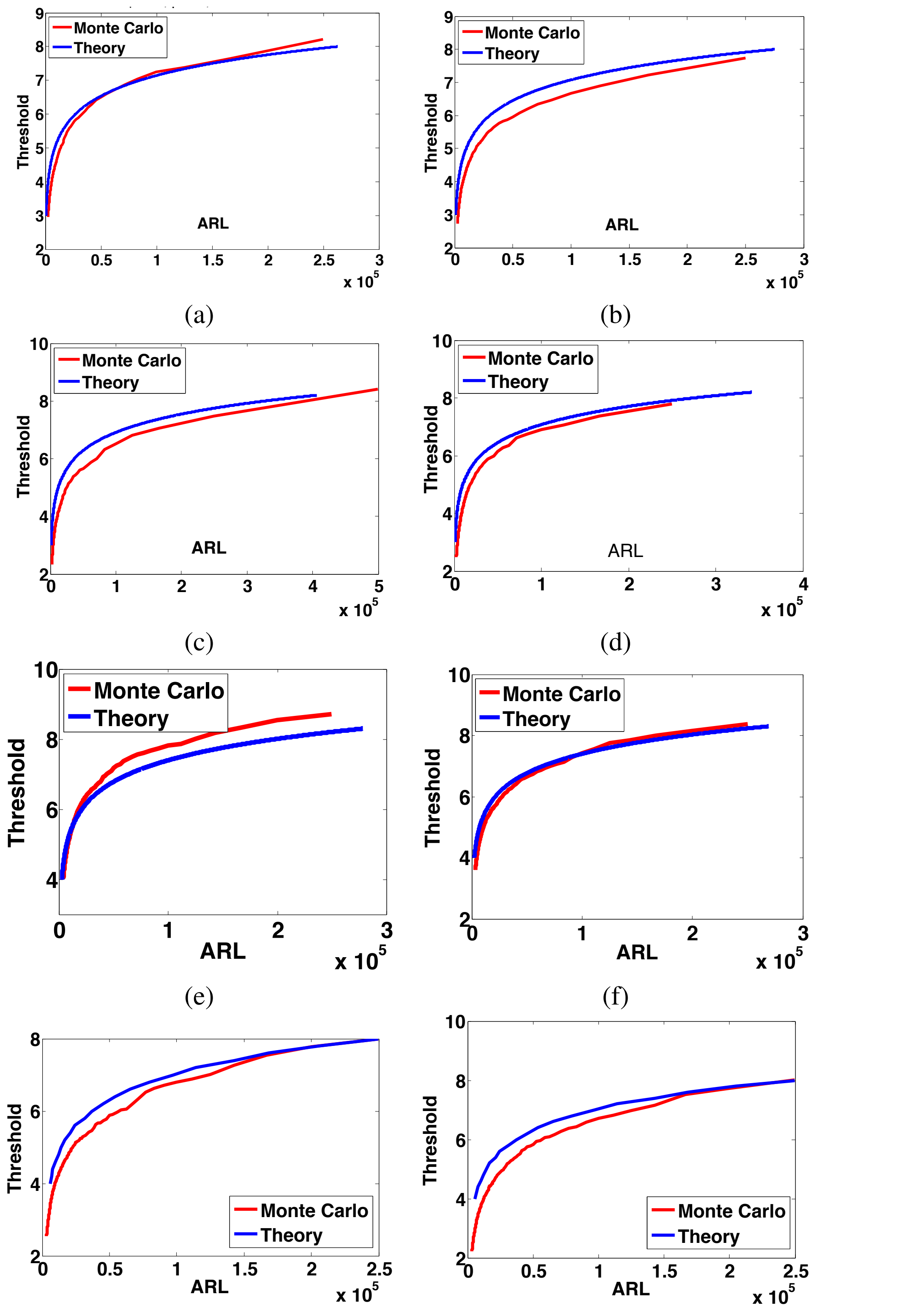
A.1 A.2 A.3

A.4 B.1 B.2

B.3 C.1 C.2

D.1 D.2 D.3

Figure 3.5: AUC curves: comparison of our method with Baseline 1.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.6: Comparison of theoretical threshold obtained via Theorem 11 with simulated threshold.

window. We set $L = 10, 50, 100$, respectively. For Fig. 3.6-(d), we select $L = 50$ and let $\beta = 10$. By comparing these four examples, we find our approximated threshold is very accurate regardless of L and β.

For Fig. 3.6-(e)(f), the null hypothesis is a one-dimensional Hawkes process with base intensity $\mu = 1$ and influence parameter $\alpha = 0.3$, $\beta = 10$. We vary the sliding window length to be $L = 100, 150$, respectively. We can see the accurate approximations as before. For Fig. 3.6-(g)(h), we consider a multi-dimensional case. The null distribution is a two dimensional Poisson processes with base intensity $\boldsymbol{\mu} = [0.5, 0.5]^\intercal$. We set $\beta = 1$ and vary the window length to be $L = 300$ and 400 respectively. The results demonstrate that our analytical threshold is also sharply accurate in the multi-dimensional situation.

3.7 Real-data

We evaluate our online detection algorithm on real Twitter and news websites data. By evaluating our log-likelihood ratio statistic on the real twittering events, we see that the statistics would rise up when there is an explanatory major event in actual scenario. By comparing the detected change points to the true major event time, we verify the accuracy and effectiveness of our proposed algorithm. In all our real experiments, we set the sliding window size to be $L = 500$ minutes, and set the kernel bandwidth β to be 1. The number of total events for the tested sequences ranges from 3000 to 15000 for every dataset.

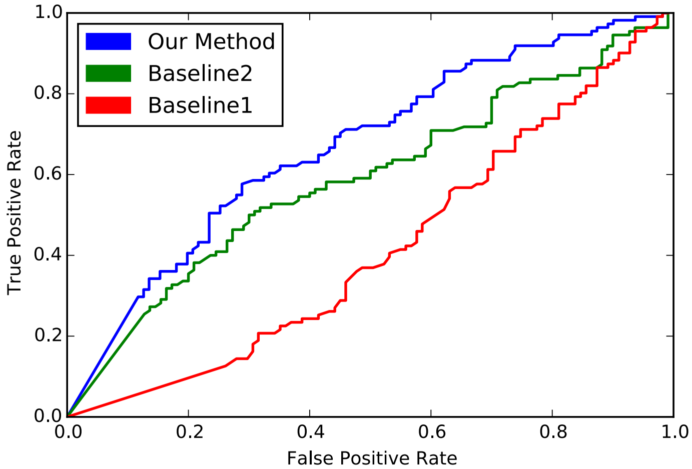


Figure 3.7: AUC for Twitter dataset on 116 important real world events.

3.7.1 Twitter Dataset

For Twitter dataset we focus on the star network topology. We create a dataset for famous people users and randomly select 30 of their followers among the tens of thousands followers. We assume there is a star-shaped network from the celebrity to the followers, and collect all their re/tweets in late January and early February 2016. Fig. 3.9-(a) demonstrates the statistics computed for the account associated to a TV series named Mr. Robot. We identify that the statistics increase around late January 10-th and early 11-th. This, surprisingly corresponds to the winning of the 2016 Golden Glob Award[1]. Fig. 3.9-(b) shows the statistics computed based on the events of the First lady of the USA and 30 of her randomly selected followers. The statistics reveal a sudden increase in 13th of January. We find a related event - Michelle Obama stole the show during the president's final State of the Union address by wearing a marigold dress which sold out even before the president finished the speech[2]. Fig. 3.9-(c) is related to Suresh Raina, an Indian professional cricketer. We selecte a small social circle around him as the center of a star-shaped network. We notice that he led his team to win an important game on Jan. 20[3], which corresponds to a sharp increase of the statistics. More results for this dataset can be found in Appendix B.5.

We further perform sensitivity analysis using the twitter data. We identify 116 important real life events. Some typical examples of such events are release of a movie/album, winning an award, Pulse Nightclub shooting, etc. Next, we identify the twitter handles associated with entities representing these events. We randomly sample 50 followers from each of these accounts and obtain a star topology graph centered around each handle. We collect tweets of all users in all these networks for a window of time before and after the real life event. For each network we compute the statistics. The AUC curves in Fig. 3.7 are obtained by varying the threshold. A threshold value is said to correctly identify the true change-point if the statistic value to the right of the change-point is greater than the

[1] http://www.tvguide.com/news/golden-globe-awards-winners-2016/

[2] http://www.cnn.com/2016/01/13/living/michelle-obama-dress-marigold-narciso-rodriguez-feat/

[3] http://www.espncricinfo.com/syed-mushtaq-ali-trophy-2015-16/content/story/963891.html

threshold. This demonstrates the good performance of our algorithm against two baseline algorithms.

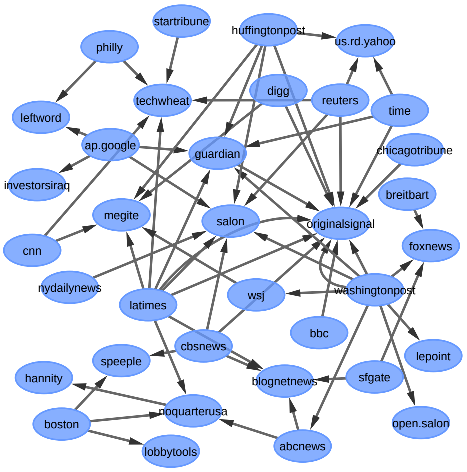


Figure 3.8: Illustration of the network topology for tracking Obama's first presidency announcement.

3.7.2 Memetracker Dataset

As a further illustration of our method, we also experiment with the Memetracker[4] dataset to detect changes in new blogs. The dataset contains the information flows captured by hyperlinks between different sites with timestamps during nine months. It tracks short units of texts and short phrases, which are called memes and act as signatures of topics and events propagation and diffuse over the web in mainstream media and blogs [80]. The dataset has been previously used in Hawkes process models of social activity [81, 57].

We create three instances of change-point detection scenarios from the Memetracker dataset using the following common procedure. First, we identify a key word associated with a piece of news occurred at κ. Second, we identify the top n websites which have the most mentions of the selected key word in a time window $[t_{\min}, t_{\max}]$ around the news break time κ (i.e., $\kappa \in [t_{\min}, t_{\max}]$). Third, we extract all articles with time stamps within $[t_{\min}, t_{\max}]$ containing the keyword, and each article is treated as an event in the point

[4] http://www.memetracker.org/

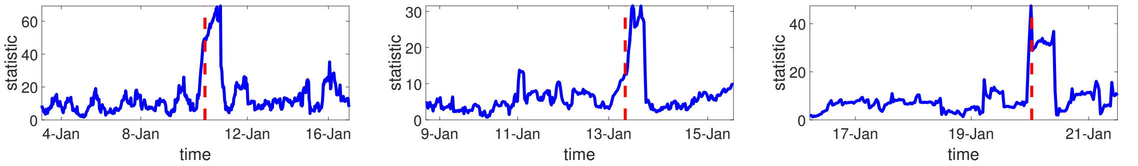


Figure 3.9: Exploratory results on Twitter for the detected change points: (left) Mr Robot wins the Golden Globe; (middle) First Lady's dress getting attention; (right) Suresh Raina makes his team won.

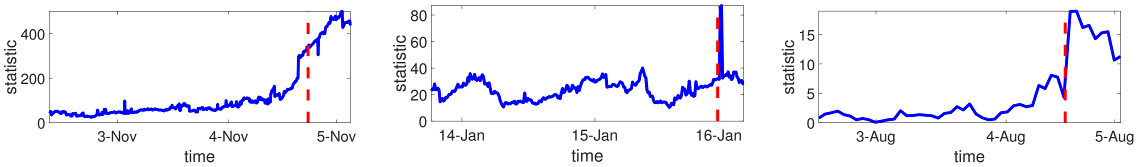


Figure 3.10: Exploratory results on Memetracker for the detected change points: (left) Obama wins the presidential election; (middle) Israel announces ceasefire; (right) Beijing Olympics starts.

process. Fourth, we construct the directed edges between the websites based on the reported linking structure. These instances correspond to real world news whose occurrences are unexpected or uncertain, and hence can cause abrupt behavior changes of the blogs. The details of these instances are showed in table 3.3.

Table 3.3: Summary information for the extracted instance for change point detection from Memetracker dataset. The keywords are highlighted in red.

real world news	n	κ	$t_{\min}$	$t_{\max}$
Obama elected president	80	11/04/08	11/02/08	11/05/08
Ceasefire in Israel	60	01/17/09	01/13/09	01/17/09
Olympics in Beijing	100	08/05/08	08/02/08	08/05/08

The first piece of news corresponds to "Barack Obama was elected as the 44th president of the United States[5]". In this example, we also plot the largest connected component of the network as shown in Fig. 3.8. It is notable that this subset includes the credible news agencies such as BBC, CNN, WSJ, Hufftingtonpost, Guardian, etc. As we show in Fig. 3.10-(a), our algorithm can successfully pinpoint a change right at the time that

[5] https://en.wikipedia.org/wiki/United_States_presidential_election,_2008

Obama was elected. The second piece of news corresponds to "the ceasefire in Israel-Palestine conflict back in 2009". Our algorithm detects a sharp change in the data, which is aligned closely to the time right before the peak of the war and one day before the Israel announces a unilateral ceasefire in the Gaza War back in 2009[6]. The third piece of news corresponds to "the summer Olympics game in Beijing". Fig. 3.10-(c) shows the evolution of our statistics. The change-point detected is 2-3 days before the opening ceremony where all the news websites started to talk about the event[7].

3.8 Summary

In this chapter, we have studied a set of likelihood ratio statistics for detecting change in a sequence of event data over networks. To the best of our knowledge, our work is the first to study change-point detection for network Hawkes process. We adopted the network Hawkes process for the event streams to model self- and mutual- excitation between nodes in the network, and cast the problem in sequential change-point detection frame, and derive the likelihood ratios under several models. We have also presented an EM-like algorithm, which can efficiently compute the likelihood ratio statistic online. The distributed nature of the algorithm enables it to be implemented on larger networks. Highly accurate theoretical approximations for the false-alarm-rate, i.e., the average-run-length (ARL) for our algorithms are derived. We demonstrated the performance gain of our algorithms relative to two baselines, which represent the current main approaches to this problem. Finally, we also tested the performance of the proposed method on synthetic and real data.

[6] http://news.bbc.co.uk/2/hi/middle_east/7835794.stm

[7] https://en.wikipedia.org/wiki/2008_Summer_Olympics

CHAPTER 4
LEARNING TEMPORAL POINT PROCESSES VIA REINFORCEMENT LEARNING

Social goods, such as healthcare, smart city, and information networks, often produce ordered event data in continuous time. The generative processes of these event data can be very complex, requiring flexible models to capture their dynamics. Temporal point processes offer an elegant framework for modeling event data without discretizing the time.

However, the existing maximum-likelihood-estimation (MLE) learning paradigm requires hand-crafting the intensity function beforehand and cannot directly monitor the goodness-of-fit of the estimated model in the process of training. To alleviate the risk of model-misspecification in MLE, we propose to generate samples from the generative model and monitor the quality of the samples in the process of training until the samples and the real data are indistinguishable.

We take inspiration from reinforcement learning (RL) and treat the generation of each event as the action taken by a stochastic policy. We parameterize the policy as a flexible recurrent neural network and gradually improve the policy to mimic the observed event distribution. Since the reward function is unknown in this setting, we uncover an analytic and nonparametric form of the reward function using an inverse reinforcement learning formulation. This new RL framework allows us to derive an efficient policy gradient algorithm for learning flexible point process models, and we show that it performs well in both synthetic and real data.

4.1 Overview

Many natural and artificial systems produce a large volume of discrete events occurring in continuous time, for example, the occurrence of crime events, earthquakes, patient visits to hospitals, financial transactions, and user behavior in mobile applications [72]. It is essential to understand and model these complex and intricate event dynamics so that accurate prediction, recommendation or intervention can be carried out subsequently depending on the context.

Temporal point processes offer an elegant mathematical framework for modeling the generative processes of these event data. Typically, parametric (or semi-parametric) assumptions are made on the intensity function [82, 83] based on prior knowledge of the processes, and the maximum-likelihood-estimation (MLE) is used to fit the model parameters from data. These models often work well when the parametric assumptions are correct. However, in many cases where the real event generative process is unknown, these parametric assumptions may be too restricted and do not reflect the reality.

Thus there emerge some recent efforts in increasing the expressiveness of the intensity function using nonparametric forms [84] and recurrent neural networks [85, 86]. However, these more sophisticated models still rely on maximizing the likelihood which now involves intractable integrals and needs to be approximated. Most recently, [87] proposed to bypass the problem of maximum likelihood by adopting a generative adversarial network (GAN) framework, where a recurrent neural network is learned to transform event sequence from a Poisson process to the target event sequence. However, this approach is rather computationally intensive, since it requires fitting another recurrent neural network as the discriminator, and it takes many iterations and careful tuning for both neural networks to reach equilibrium.

In this chaper, we take a new perspective and establish an under-explored connection between temporal point processes and reinforcement learning: the generation of each event can be treated as the action taken by a stochastic policy, and the intensity function learning problem in temporal point processes can be viewed as the policy learning problem in reinforcement learning.

More specifically, we parameterize a stochastic policy π using a recurrent neural net-

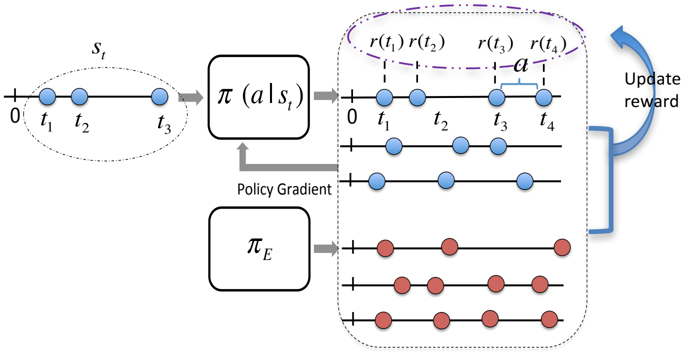


Figure 4.1: Illustration of our RL modeling framework.

work over event history and learn the unknown reward function via *inverse reinforcement learning* [91, 92, 93, 94]. Our algorithm for policy optimization iterates between learning the reward function and the stochastic policy π. Inverse reinforcement learning is known to be time-consuming, which requires solving a reinforcement learning problem in every inner-loop. To tackle this problem, we convert the inverse reinforcement learning step to a minimization problem over the discrepancy between the expert point process and the learner point process. By choosing the function class of reward to be the unit ball in reproducing kernel Hilbert space (RKHS) [95, 96, 97], we can get an explicit nonparametric closed form for the optimal reward function. Then the stochastic policy can be learned by a customized policy gradient with the optimal reward function having an analytical expression.

An illustration of our modeling framework is shown in Figure 4.1. The observed trajectories of events will be viewed as the actions generated by an *expert* policy π_E. The goal is to learn a policy which we call *learner* that mimics the distribution of the observed expert event sequences. The learner policy $\pi(a|s_t)$ provides the probability of the next event occurring at a after t, and $s_t := \{t_i\}_{t_i<t}$ is the history of events before t. We parametrize $\pi(a|s_t)$ by a recurrent neural network (RNN) with stochastic neurons [88], where the generated events are fed back to the RNN leading to a double stochastic point process [89]. Furthermore, each generated event t_i will be also associated with a reward $r(t_i)$, and the policy will be learned by maximizing the expected cumulative rewards [90].

We conducted experiments on various synthetic and real sequences of event data and showed that our approach outperforms the state-of-the-art regarding both data description and computational efficiency.

4.2 Preliminaries

4.2.1 Temporal Point Processes.

A temporal point process is a stochastic process whose realization is a sequence of discrete events $\{t_i\}$ with $t_i \in \mathbb{R}^+$ and $i \in \mathbb{Z}^+$ abstracted as points on a timeline [72]. Let the history

$$\boldsymbol{s}_t = \{t_1, t_2, \ldots, t_n | t_n < t\}$$

be the sequence of event times up to but not including time t. The intensity function (rate function) $\lambda(t|\boldsymbol{s}_t)$ conditioned on the history $\boldsymbol{s}_t$ uniquely characterizes the generative process of the events. Different functional forms of $\lambda(t|\boldsymbol{s}_t)dt$ capture different generating patterns of events. For example, a plain homogeneous Poisson process has

$$\lambda(t|\boldsymbol{s}_t) = \lambda_0 \geqslant 0,$$

implying that each event occurs independently of each and uniformly on the timeline. A Hawkes process has

$$\lambda(t|\boldsymbol{s}_t) = \lambda_0 + \sum_{t_i \in s_t} \exp(-(t - t_i))$$

where the occurrences of past events will boost future occurrences. Given the intensity function, the survival function defined as

$$S(t|\boldsymbol{s}_t) = \exp(-\int_{t_n}^{t} \lambda(\tau) d\tau)$$

is the conditional probability that no event occurs in the window $[t_n, t)$, and the likelihood of observing event at time t is defined as

$$f(t|\boldsymbol{s}_t) = \lambda(t|\boldsymbol{s}_t)S(t|\boldsymbol{s}_t)$$

. Then we can express the joint likelihood of observing a sequence of events $\boldsymbol{s}_T = \{t_1, t_2, \ldots, t_n | t_n < T\}$ up to an observation window T as

$$p(\{t_1, t_2, \ldots, t_n | t_n < T\}) = \prod_{t_i \in \boldsymbol{s}_T} \lambda(t_i|\boldsymbol{s}_{t_i}) \cdot \exp\left(-\int_0^T \lambda(\tau|\boldsymbol{s}_\tau)d\tau\right). \tag{4.1}$$

The integral normalization in the likelihood function can be intensive to compute especially in cases where $\lambda(t|\boldsymbol{s}_t)$ do not have a simple form. In this case, a numerical approximation is typically needed which may affect the accuracy of the fitting process.

4.2.2 Reproducing Kernel Hilbert Spaces.

A reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ on $\mathcal{T}$ with a kernel $k(t, t')$ is a Hilbert space of functions $f(\cdot) : \mathcal{T} \mapsto \mathbb{R}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$. Its element $k(t, \cdot)$ satisfies the reproducing property:

$$\langle f(\cdot), k(t, \cdot) \rangle_{\mathcal{H}} = f(t),$$

and consequently,

$$\langle k(t, \cdot), k(t', \cdot) \rangle_{\mathcal{H}} = k(t, t')$$

meaning that we can view the evaluation of a function f at any point $t \in \mathcal{T}$ as an inner product. Commonly used RKHS kernel function includes Gaussian radial basis function (RBF) kernel

$$k(t, t') = \exp(-\|t - t'\|^2 / 2\sigma^2)$$

where $\sigma > 0$ is the kernel bandwidth, and polynomial kernel

$$k(t,t') = (\langle t,t'\rangle + a)^d$$

where $a > 0$ and $d \in \mathbb{N}$ [31, 98, 95]. In this paper, if not otherwise stated, we will assume that Gaussian RBF kernel is used. Let $\mathbb{P}$ be a measure on $\mathcal{T}$, we define the mapping of $\mathbb{P}$ to RKHS,

$$\mu_{\mathbb{P}} := \mathbb{E}_{\mathbb{P}}[k(t,\cdot)] = \int_{t\in\mathcal{T}} k(t,\cdot)\,\mathrm{d}\mathbb{P}(t),$$

as the Hilbert space embedding of $\mathbb{P}$ [99]. Then for all $f \in \mathcal{H}$,

$$\mathbb{E}_{\mathbb{P}}[f(t)] = \langle f, \mu_{\mathbb{P}}\rangle_{\mathcal{H}}$$

by the reproducing property. Similarly, one can also embed another measure $\mathbb{Q}$ on $\mathcal{T}$ into RKHS as $\mu_{\mathbb{Q}}$. Then a distance between measure $\mathbb{P}$ and $\mathbb{Q}$ can be defined as

$$\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} := \sup_{\|f\|_{\mathcal{H}}\leqslant 1} \langle f, \mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\rangle_{\mathcal{H}}\,.$$

A characteristic RKHS is one for which the embedding is injective: that is, each measure has a unique embedding [100], and

$$\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} = 0$$

if and only if $\mathbb{P} = \mathbb{Q}$. This property holds for many commonly used kernels. For $\mathcal{T} = \mathbb{R}^d$, this includes the Gaussian kernels.

4.3 A Reinforcement Learning Framework

Suppose we are interested in modeling the daily crime patterns, or monthly occurrences of disease for patients, then the data are collected as trajectories of events within a predefined time window T. We regard the observed paths as actions taken by an expert (nature).

Let $\xi = \{\tau_1, \tau_2, \dots, \tau_{N_T^\xi}\}$ represent a single trajectory of events from the expert where N_T^ξ is the total number of events up to T, and it can be different for different sequences. Then, each trajectory $\xi \sim \pi_E$ can be seen as an expert demonstration sampled from the expert policy π_E. Hence, on a high level, given a set of expert demonstrations

$$\mathcal{D} = \{\xi_1, \xi_2, \dots, \xi_j, \dots | \xi_j \sim \pi_E\},$$

we can treat fitting a temporal point process to $\mathcal{D}$ as searching for a learner policy π_θ which can generate another set of sequences

$$\tilde{\mathcal{D}} = \{\eta_1, \eta_2, \dots, \eta_j, \dots | \eta_j \sim \pi_\theta\}$$

with similar patterns as $\mathcal{D}$. We will elaborate on this reinforcement learning framework below.

4.3.1 Reinforcement Learning Formulation (RL).

Given a sequence of past events $\boldsymbol{s}_t = \{t_i\}_{t_i<t}$, the stochastic policy $\pi_\theta(a|\boldsymbol{s}_t)$ samples an inter-event time a as its action to generate the next event time as $t_{i+1} = t_i + a$. Then, a reward $r(t_{i+1})$ is provided and the state $\boldsymbol{s}_t$ will be updated to $\boldsymbol{s}_t = \{t_1, \dots, t_i, t_{i+1}\}$. Fundamentally, the policy $\pi_\theta(a|\boldsymbol{s}_t)$ corresponds to the conditional probability of the next event time in temporal point process, which in turn uniquely determines the corresponding intensity function as

$$\lambda_\theta(t|\boldsymbol{s}_{t_i}) = \frac{\pi_\theta(t - t_i|\boldsymbol{s}_{t_i})}{1 - \int_{t_i}^{t} \pi_\theta(\tau - t_i|\boldsymbol{s}_{t_i}) d\tau}.$$

This builds the connection between the intensity function in temporal point processes and the stochastic policy in reinforcement learning. If reward function $r(t)$ is given, the optimal policy π_θ^* can be directly computed via

$$\pi_\theta^* = \arg \max_{\pi_\theta \in \mathcal{G}} \;\; J(\pi_\theta) := \mathbb{E}_{\eta \sim \pi_\theta} \left[\sum\nolimits_{i=1}^{N_T^\eta} r(t_i) \right], \tag{4.2}$$

where $\mathcal{G}$ is the family of all candidate policies π_θ, $\eta = \{t_1, \ldots, t_{N_T^\eta}\}$ is one sampled roll-out from policy π_θ, and N_T^η can be different for different roll-out samples.

4.3.2 Inverse Reinforcement Learning (IRL).

Eq.(4.2) shows that when the reward function is given, the optimal policy can be determined by maximizing the expected cumulative reward. However, in our case, only the expert's sequences of events can be observed, but the real reward function is unknown. Given the expert policy π_E, IRL can help to uncover the optimal reward function $r^*(t)$ by

$$r^* = \max_{r \in \mathcal{F}} \left(\mathbb{E}_{\xi \sim \pi_E} \left[\sum\nolimits_{i=1}^{N_T^\xi} r(\tau_i) \right] - \max_{\pi_\theta \in \mathcal{G}} \mathbb{E}_{\eta \sim \pi_\theta} \left[\sum\nolimits_{i=1}^{N_T^\eta} r(t_i) \right] \right), \tag{4.3}$$

where $\mathcal{F}$ is the family class for reward function, $\xi = \{\tau_1, \ldots, \tau_{N_T^\xi}\}$ is one event sequence generated by the expert π_E, and $\eta = \{t_1, \ldots, t_{N_T^\eta}\}$ is one roll-out sequence from the learner π_θ. The formulation means that a proper reward function should give the expert policy higher reward than any other learner policy in $\mathcal{G}$, and thus the learner can approach the expert performance by maximizing this reward. Denote the procedure (4.2) and (4.3) as RL(r) and IRL(π_E), accordingly. The optimal policy can be obtained by

$$\pi_\theta^* = \text{RL} \circ \text{IRL}(\pi_E). \tag{4.4}$$

4.3.3 Overview of the Proposed Learning Framework.

Solving the optimization problem (4.3) is very time-consuming in that it requires to solve the inner loop RL problem repeatedly. We relieve the computational challenge by choosing the space of functions $\mathcal{F}$ for $r(t)$ to be the unit ball in RKHS $\mathcal{H}$, which allows us to obtain an analytical expression for the updated reward function $\hat{r}(t)$ given any current learner policy $\hat{\pi}(\theta)$. This $\hat{r}(t)$ is determined by finite sample expert trajectories and finite sample roll-outs from the current learner policy, and it directly quantifies the discrepancy between the expert's policy (or intensity function) and current learner policy (or intensity function). Then by solving a simple RL problem as in (4.2), the learner policy can be improved to close its gap with the expert policy using a simple policy gradient type of algorithm.

4.4 Model

In this section, we present model parametrization and the analytical expression of optimal reward function.

4.4.1 Policy Network.

The function class of the policy $\pi_\theta \in \mathcal{G}$ should be flexible and expressive enough to capture the potential complex point process patterns of the expert. We, therefore, adopt the recurrent neural network (RNN) with stochastic neurons [88] which is flexible to capture the nonlinear and long-range sequential dependency structure. More specifically,

$$a_i \sim \pi(a \,|\, \Theta(h_{i-1})), \;\; h_i = \psi(Va_i + Wh_{i-1}), \;\; h_0 = 0, \tag{4.5}$$

where the hidden state $h_i \in \mathbb{R}^d$ encodes the sequence of past events $\{t_1, \ldots, t_i\}$, $a_i \in \mathbb{R}^+$, $V \in \mathbb{R}^d$, and $W \in \mathbb{R}^{d\times d}$. Here ψ is a nonlinear activation function applied element-wise, and Θ is a nonlinear mapping from $\mathbb{R}^d$ to the parameter space of the probability distribution π. For instance, one can choose $\psi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ to be the tanh function, and design the output layer of Θ such that $\Theta(h_{i-1})$ is a valid parameter for a probability density function π. The output $a_i = t_i - t_{i-1}$, serves as the i-th inter-event time (let $t_0 = 0$), and $a_i > 0$. The choice of model π is quite flexible, only with the constraint that the random variable should be positive since a is always positive. Common distributions such as exponential and Rayleigh distributions would satisfy such constraint, leading to $\pi(a|\Theta(h_{i-1})) = \Theta(h)e^{-\Theta(h)a}$ and $\pi(a|\Theta(h_{i-1})) = \Theta(h)ae^{-\Theta(h)a^2/2}$ respectively. In this way, we specify a nonlinear and flexible dependency over the history.

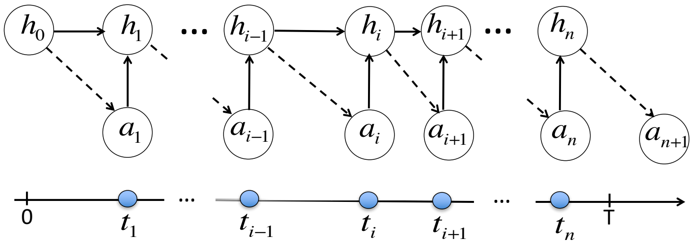


Figure 4.2: Illustration of generator π_θ.

The architecture of our model in (4.5) is shown in Figure 4.2. Different from traditional RNN, the outputs a_i are sampled from π rather than obtained by deterministic transformations. This is what “stochastic” policy means. Randomly sampling will allow the policy to explore the temporary space. Furthermore, the sampled time point will be fed back to the RNN. The proposed model aims to capture that the state h_i is attributed by two parts. One is the *deterministic* influence from the previous hidden state h_{i-1}, and the other is the *stochastic* influence from the latest sampled action a_i. Action a_i is sampled from the previous distribution $\pi(a|\Theta(h_{i-1}))$ with parameter $\Theta(h_{i-1})$ and will be fed back to influence the current hidden state h_i.

In some sense, our RNN with stochastic neurons mimics the event generating mechanism of the doubly stochastic point process, such as Hawkes process and self-correcting process. For these types of point processes, the intensity is stochastic, which depends on history, and the intensity function will control the occurrence rate of the next event.

4.4.2 Reward Function Class.

The reward function directly quantifies the discrepancy between π_E and π_θ, and it guides the learning of the optimal policy π_θ^*. On the one hand, we want its function class $r \in \mathcal{F}$ to be sufficiently flexible so that it can represent the reward function of various shapes. On the other hand, it should be restrictive enough to be efficiently learned with finite samples [98, 95]. With these competing considerations, we choose $\mathcal{F}$ to be the unit ball in RKHS $\mathcal{H}$, $\|r\|_{\mathcal{H}} \leqslant 1$. An immediate benefit of this function class is that we can show the optimal policy can be directly learned via a minimization formulation given in Theorem 14 instead of the original minimax formulation (4.3).

A sketch of proof is provided as follows. For short notation, we denote

$$\underbrace{\phi(\eta) := \int_{[0,T)} k(t,\cdot) dN_t^{(\eta)},}_{\text{feature mapping from data space to R}} \quad \text{and} \quad \underbrace{\mu_{\pi_\theta} := \mathbb{E}_{\eta\sim\pi_\theta}\left[\phi(\eta)\right]}_{\text{mean embeddings of the intensity function in RKHS}}$$

where $dN_t^{(\eta)}$ is the counting process associated with sample path η, and $k(t,t')$ is a universal RKHS kernel. Then using the reproducing property,

$$J(\pi_\theta) := \mathbb{E}_{\eta\sim\pi_\theta}\left[\sum_{i=1}^{N_T^{(\eta)}} r(t_i)\right] = \mathbb{E}_{\eta\sim\pi_\theta}\left[\int_{[0,T)} \langle r, k\left(t,\cdot\right)\rangle_{\mathcal{H}} dN_t^{(\eta)}\right] = \langle r, \mu_{\pi_\theta}\rangle_{\mathcal{H}}.$$

Similarly, we can obtain $J(\pi_E) = \langle r, \mu_{\pi_E}\rangle_{\mathcal{H}}$. From (4.3), r^* is obtained by

$$\max_{\|r\|_{\mathcal{H}}\leq 1} \min_{\pi_\theta\in\mathcal{G}} \langle r, \mu_{\pi_E} - \mu_{\pi_\theta}\rangle_{\mathcal{H}} = \min_{\pi_\theta\in\mathcal{G}} \max_{\|r\|_{\mathcal{H}}\leq 1} \langle r, \mu_{\pi_E} - \mu_{\pi_\theta}\rangle_{\mathcal{H}} = \min_{\pi_\theta\in\mathcal{G}} \|\mu_{\pi_E} - \mu_{\pi_\theta}\|_{\mathcal{H}},$$

where the first equality is guaranteed by the minimax theorem, and

$$r^*(\cdot|\pi_E,\pi_\theta) = \frac{\mu_{\pi_E} - \mu_{\pi_\theta}}{\|\mu_{\pi_E} - \mu_{\pi_\theta}\|_{\mathcal{H}}} \propto \mu_{\pi_E} - \mu_{\pi_\theta} \tag{4.6}$$

can be empirically evaluated by data. In this way, we change the original minimax formulation for solving π_θ^* to a simple **minimization** problem, which will be more efficient and stable to solve in practice. We summarize the formulation in Theorem 14.

Theorem 14 *Let the family of reward function be the unit ball in RKHS $\mathcal{H}$, i.e., $\|r\|_{\mathcal{H}} \leqslant 1$.*

Then the optimal policy obtained by (4.4) can also be obtained by solving

$$\pi_\theta^* = \arg\min_{\pi_\theta \in \mathcal{G}} D(\pi_E, \pi_\theta, \mathcal{H}) \tag{4.7}$$

where $D(\pi_E, \pi_\theta, \mathcal{H})$ is the maximum expected cumulative reward discrepancy between π_E and π_θ,

$$D(\pi_E, \pi_\theta, \mathcal{H}) := \max_{\|r\|_{\mathcal{H}} \leqslant 1} \left(\mathbb{E}_{\xi \sim \pi_E} \left[\sum\nolimits_{i=1}^{N_T^{(\xi)}} r(\tau_i) \right] - \mathbb{E}_{\eta \sim \pi_\theta} \left[\sum\nolimits_{i=1}^{N_T^{(\eta)}} r(t_i) \right] \right). \tag{4.8}$$

Theorem 14 implies that we can transform the inverse reinforcement learning procedure of (4.4) to a simple minimization problem which minimizes the maximum expected cumulative reward discrepancy between π_E and π_θ. This enables us to sidestep the expensive computation of (4.4) caused by the solving the inner RL problem repeatedly. What's more interesting, we can derive an analytical solution to (4.8) given by (4.6).

4.4.3 Finite Sample Estimation.

Given L trajectories of expert point processes, and M trajectories of events generated by π_θ, mean embeddings μ_{π_E} and μ_{π_θ} can be estimated by their respective empirical mean: $\hat{\mu}_{\pi_E} = \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{N_T^{(l)}} k(\tau_i^{(l)}, \cdot)$ and $\hat{\mu}_{\pi_\theta} = \frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{N_T^{(m)}} k(t_i^{(m)}, \cdot)$. Then for any $t \in [0, T)$, the estimated optimal reward is (without normalization) is

$$\hat{r}^*(t) \propto \frac{1}{L}\sum\nolimits_{l=1}^{L}\sum\nolimits_{i=1}^{N_T^{(l)}} k(\tau_i^{(l)}, t) - \frac{1}{M}\sum\nolimits_{m=1}^{M}\sum\nolimits_{i=1}^{N_T^{(m)}} k(t_i^{(m)}, t). \tag{4.9}$$

Note this empirical estimator is biased at $\tau_i^{(l)}$ and $t_i^{(m)}$. Unbiased estimator can also be obtained and will be provided in **Algorithm RLPP** discussed later for simplicity.

4.4.4 Kernel Choice.

The unit ball in RKHS is dense and expressive. Fundamentally, our proposed framework and theoretical results are general and can be directly applied to other types of kernels. For example, we can use the Matérn kernel, which generates spaces of differentiable functions known as the Sobolev spaces [101, 102]. In later experiments, we have used Gaussian kernel and obtained promising results.

4.5 Learning Algorithm

4.5.1 Learning via Policy Gradient.

In practice, instead of minimizing $D(\pi_E, \pi_\theta, \mathcal{H})$ as in (4.7), we can equivalently minimize $D(\pi_E, \pi_\theta, \mathcal{H})^2$ since square is a monotonic transformation. Now, we can learn π_θ^* from the RL formulation (4.2) using policy gradient with variance reduction. First, with the likelihood ratio trick, the gradient of $\nabla_\theta D(\pi_E, \pi_\theta, \mathcal{H})^2$ can be computed as

$$\nabla_\theta D(\pi_E, \pi_\theta, \mathcal{H})^2 = \mathbb{E}_{\eta\sim\pi_\theta}\left[\sum_{i=1}^{N_T^\eta}\left(\nabla_\theta \log \pi_\theta(a_i|\Theta(h_{i-1}))\right)\cdot\left(\sum\nolimits_{i=1}^{N_T^\eta}\hat{r}^*(t_i)\right)\right], \qquad (4.10)$$

where $\sum_{i=1}^{N_T^\eta}\left(\nabla_\theta \log \pi_\theta(a_i|\Theta(h_{i-1}))\right)$ is the gradient of the log-likelihood of a roll-out sample $\eta = \{t_1, \ldots, t_{N_T^\eta}\}$ using the learner policy π_θ.

To reduce the variance of the gradient, we can exploit the observation that future actions do not depend on past rewards. This leads to a variance reduced gradient estimate

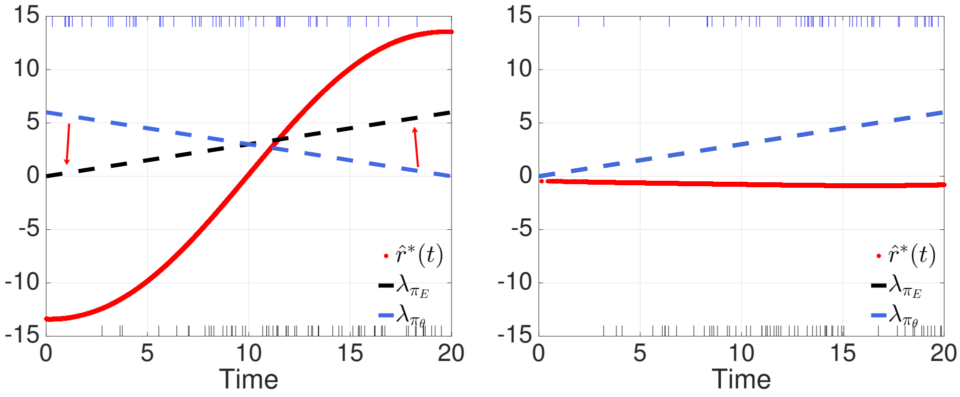
$$\nabla_\theta D(\pi_E, \pi_\theta, \mathcal{H})^2 = \mathbb{E}_{\eta\sim\pi_\theta}\left[\sum_{i=1}^{N_T^\eta}\left(\nabla_\theta \log \pi_\theta(a_i|\Theta(h_{i-1}))\right)\cdot\left(\sum_{l=i}^{N_T^\eta}\left[\hat{r}^*(t_l) - b_l\right]\right)\right]$$

where $\left(\sum_{l=i}^{N_T}\hat{r}^*(t_l)\right)$ is referred to as the "reward to go" and b_l is the baseline to further reduce the variance. The overall procedure is given in **Algorithm RLPP**. In the algorithm, after we sample M trajectories from the current policy, we use one trajectory η^m for evaluation and the rest $M-1$ samples to estimate reward function.

An example reward function learned at a different stage of the algorithm is also illustrated in Figure 4.3. The reward function $\hat{r}^*(t)$ is estimated using 100 sampled sequences from π_E and π_θ. In (a), $\hat{r}^*(t) > 0$ when the expert's intensity is above the learner's intensity, and $\hat{r}^*(t) < 0$ when the expert's intensity is below the learner's intensity. In order to maximize the cumulative reward given the current reward, the learner should generate more events in the region when $\hat{r}^*(t) > 0$ and reduce the number of events when $\hat{r}^*(t) < 0$. Based on our formulation, the optimal reward function always quantifies the discrepancy between the expert and current learner by considering the worst case. As a result, once the learner is changed, the current optimal reward $\hat{r}^*(t)$ is updated accordingly, and $\hat{r}^*(t)$ guides the learner to update its policy towards mimicking the expert's behavior until they exactly match each other in (b) where $\hat{r}^*(t)$ becomes zero.

Algorithm 2 RLPP: Mini-batch Reinforcement Learning for Learning Point Processes

```
Initialize model parameters $\theta$
**for** number of training iterations **do**
    • Sample minibatch of $L$ trajectories of events $\{\xi^{(1)}, \dots, \xi^{(L)}\}$ from expert, where $\xi^{(l)} = \{\tau_1^{(l)}, \dots, \tau_{N_T^{(l)}}^{(l)}\}$
    • Sample minibatch of $M$ trajectories of events $\{\eta^{(1)}, \dots, \eta^{(M)}\}$ from policy $\pi_\theta(a|s)$, where $\eta^{(m)} = \{t_1^{(m)}, \dots, t_{N_T}^{(m)}\}$
    • Update $\pi_\theta$ by policy gradient:
        $$\theta \leftarrow \theta + \alpha \nabla_\theta \frac{1}{M} \sum_{m=1}^{M} \left( \sum_{i=1}^{N_T^{(m)}} \hat{r}^*(t_i^{(m)}) \log p_\theta(\eta^{(m)}) \right)$$
    where $\log p_\theta(\eta^{(m)}) = \sum_{i=1}^{N_T^\eta} (\log \pi_\theta(a_i|\Theta(h_{i-1})))$ is the log-likelihood of the sample $\eta^{(m)}$, and $r^*(t_i^{(m)})$ can be estimated by $L$ expert trajectories and $(M-1)$ roll-out samples without $\eta^{(m)}$
        $$\hat{r}^*(t) = \frac{1}{L} \sum_{l=1}^{L} \sum_{i=1}^{N_T^{(l)}} k(\tau_i^{(l)}, t) - \frac{1}{M-1} \sum_{m'=1, m' \neq m}^{M} \sum_{j=1}^{N_T^{(m')}} k(t_j^{(m')}, t).$$
**end for**
```



(a)

(b)

Figure 4.3: Generated events v.s. training events and the estimate reward function $\hat{r}^*(t)$.

4.5.2 Comparison with MLE.

During training, our generative model directly compares the generated temporal events with the observed events to iteratively correct the mistakes, which can effectively avoid model misspecification. Since the training only involves the policy gradient, it bypasses the intractability issue of the log-survival term in the likelihood (Eq. (4.1)). On the other hand, because the learned policy is in fact the conditional density of a point process, our approach still resembles the form of MLE in the RL reformulation and can thus be interpreted in a statistically principled way.

4.5.3 Comparison with GAN and GAIL.

By Theorem 14, our policy is learned directly by minimizing the discrepancy between π_E and π_θ which has a closed form expression. Thus, we convert the original IRL problem to a minimization problem with only one set of parameters with respect to the policy. In each training iteration with the policy gradient, we have an unbiased estimator of the gradient, and the estimated reward function also depends on the current policy π_θ. In contrast, in GAN or GAIL formulation, they have two sets of parameters related to the generator and the discriminator. The gradient estimator is biased because each min-/max-problem is in fact nonconvex and cannot be solved in one-shot. Thus, our framework is more stable and efficient than the mini-max formulation for learning point processes.

4.6 Experiments

4.6.1 Synthetic datasets.

To show the robustness to model-misspecifications of our approach, we propose the following four different point processes as the ground-truth: (I) **Inhomogeneous Poisson (IP)** with $\lambda(t) = at + b$ where $a = -0.2$ and $b = 3.5$; Here we omit $\boldsymbol{s}_t$ since $\lambda(t)$ does not depend on the history. (II) **Hawkes Process (HP)** with $\lambda(t|\boldsymbol{s}_t) = \mu + \alpha \sum_{t_i<t} \exp\{-(t-t_i)\}$ where $\mu = 2$, and $\alpha = 0.5$. (III) Mixture of IP and HP version 1 (**IP + HP1**). For the IP component, its $\lambda(t)$ is piece-wise linear with monotonic increasing slopes of pieces from $\{0.2, 0.3, 0.4, 0.5\}$. The HP component has the parameter $\mu = 1$ and $\alpha = 0.5$; (IV) Mixture of IP and HP version 2 (**IP + HP2**) where the IP component also has piece-wise linear intensity but the slopes have the zig-zag pattern chosen from $\{1, -1, 2, -2\}$, and the HP component has the parameter $\mu = 1$ and $\alpha = 0.1$.

4.6.2 Real datasets.

We evaluate our approach on four real datasets across a diverse range of domains:

- **911 call dataset** contains 220,000 crime incident call records from 2011 to 2017 in Atlanta area. We select one beat zone data with call timestamps ranging from 7:00 AM to 1:00 PM.

- Microsoft Academic Search (**MAS**) provides access to publication venues, time, citations, etc. We collect citation records for 50,000 papers and treat each citation time as an event.

- Medical Information Mart for Intensive Care III (**MIMIC-III**) contains de-identified clinical visit records from 2001 to 2012 for more than 40,000 patients. Our data contain 2,246 patients with at least 3 visits. For a given patient, each clinical visit will be treated as an event.

- **NYSE** contains 0.7 million high-frequency trading records from NYSE for a given stock within one day. All transactions are evenly divided into 3,200 segments. All segments have the same temporal duration. Each trading record is treated as a event.

4.6.3 Baselines.

We compare our approach against two state-of-the-arts as well as conventional parametric baselines. The two state-of-the-art methods are WGANTPP [87] and RMTPP[1] [85]. In addition, three parametric methods based on maximum likelihood estimation are compared, including: (1) Inhomogeneous Poisson process where the intensity function is modeled using a mixture of Gaussian components, (2) Hawkes Process (or Self-Excitation process denoted as **SE**), and (3) Self-Correcting process (**SC**) with $\lambda(t|\boldsymbol{s}_t) = \exp\left\{\mu t - \sum_{t_i<t} \alpha\right\}$. In contrast to Hawkes process, the self-correcting process seeks to produce regular point patterns. The intuition is that while the intensity increases steadily, every time when a new event appears, it is decreased by multiplying a constant $e^{-\alpha} < 1$, so the chance of new points decreases after an event has occurred recently.

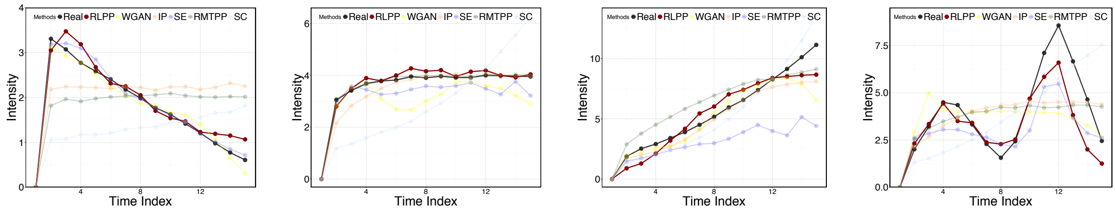
4.6.4 Experimental Setup.

The policy in our method RLPP is parameterized as LSTM with 64 hidden neurons, and $\pi(a|\Theta(h))$ is chosen to be exponential distribution. Batch size is 32 (the number of sampled sequences L and M are 32 in Algorithm 1, and learning rate is 1e-3. We use Gaussian kernel $k(t, t') = \exp(-\|t - t'\|^2/\sigma^2)$ for the reward function. The kernel bandwidth σ is estimated using the "median trick" based on the observations [95]. For WGANTPP and RMTPP, we are using the open source codes. For WGANTPP[2], we have used the exact

[1] RMTPP has very similar performance with [86].

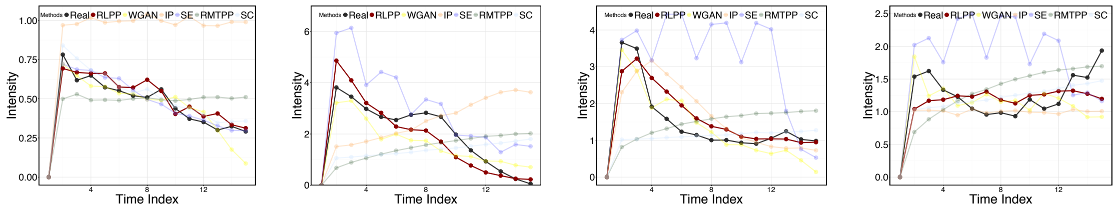
[2] https://github.com/xiaoshuai09/Wasserstein-Learning-For-Point-Process

experimental setup as [87], which adopts Adam optimization method [103] with learning rate 1e-4, $\beta_1 = 0.5$, $\beta_2 = 0.9$, and the batch size is 256. For RMTPP[3], batch size is 256, state size is 64, and learning rate is 1e-4.



IP HP IP + HP1 IP + HP2

Figure 4.4: Comparison of empirical intensity functions on the synthetic data.



911 Call MAS MIMIC III NYSE

Figure 4.5: Comparison of empirical intensity functions on the real datasets.

4.6.5 Comparison of Learned Empirical Intensity.

We first compare the empirical intensity of the learner point process to the expert point process. This is a straightforward comparison: one can visually assess the performance and localize the discrepancy. Fig. 4.4 and Fig. 4.5 demonstrate the empirical intensity functions of generated sequences based on synthetic and real data. For each dataset, we have used all learned models to generate new sequences. The comparisons are based on the empirical intensities estimated from the generated temporal events and those estimated from the observed temporal events. It clearly shows that RLPP consistently outperforms RMTPP, and achieves comparable and sometimes even better fitting against WGANTPP. Furthermore, RLPP consistently outperforms the other three conventional parametric models when there

[3] https://github.com/dunan/NeuralPointProcess

exist model-misspecifications. Without any prior knowledge, RLPP can capture the major trends in data and can accurately learn the nonlinear dependency structure hidden in data. In the Hawkes example, RLPP performs even as accurate as the ground-truth model. On the real-world data, the underlying true model is unknown and the point process patterns are more complicated. RLPP still shows a decent performance in the real datasets.

4.6.6 Comparison of Data Fitting.

Quantile plot (QQ-plot) for residual analysis is a standard model checking approach for general point processes. Given a set of real input samples $t_1, \dots, t_n$, by the Time Changing Theorem [72], if such set of samples is one realization of a process with the intensity $\lambda(t)$, then the respective value achieved from the integral $\Lambda = \int_{t_{i-1}}^{t_i} \lambda(t)dt$ should conform to the unit-rate exponential distribution [104]. For the synthetic experiments, since we know the exact ground-truth parametric form of $\lambda(t|\boldsymbol{s}_t)$, we can perform this explicit transformation for a test. Ideally, the QQ-plot for the generated sequences should follow a 45-degree straight line. We use Hawkes Process (HP) and Inhomogeneous Poisson Process + Hawkes Process (IP+HP1) dataset to produce the QQ-plot and compare different methods in Fig. 4.6. In both cases, RLPP consistently stands out even without any prior knowledge about the parametric form of the true underlying generative point process and the fitting slope is very close to the diagonal line in both cases. More rigorously, we perform the KS test. Fig. 4.7 illustrates the cumulative distributions (CDF) of p-values. We followed the experiment setup in [105]: we generated samples from each learned point process models, transformed the time interval, and applied the KS test to compare with unit rate exponential distribution. Under this null hypothesis, the distribution of the p-values over tests should follow a uniform distribution, whose CDF should be a diagonal line. If the target distribution is the Hawkes process (Fig. 4.7), both the learned SE (Hawkes process) and the RLPP models are indistinguishable from that.

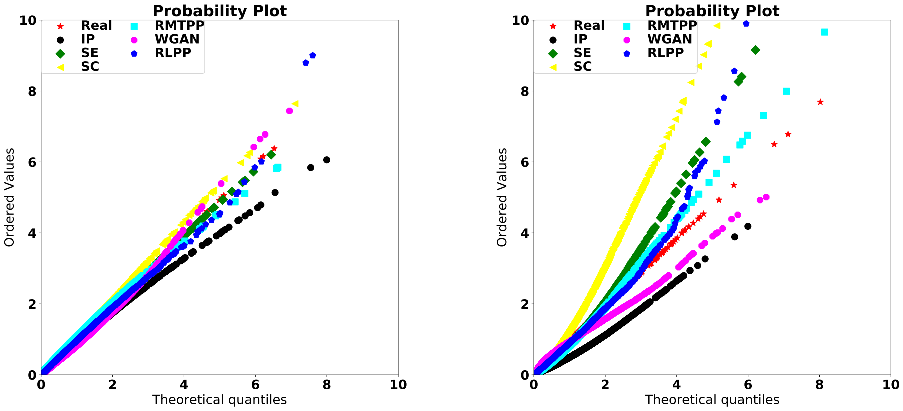


Figure 4.6: QQ-plot for dataset HP (left) and HP+IP1 (right).

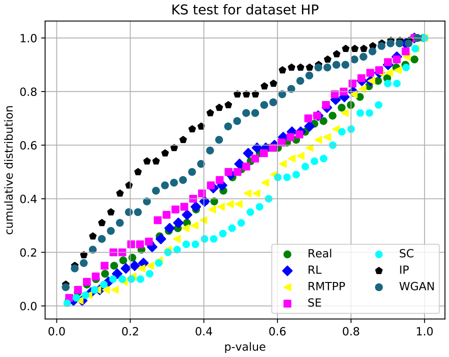


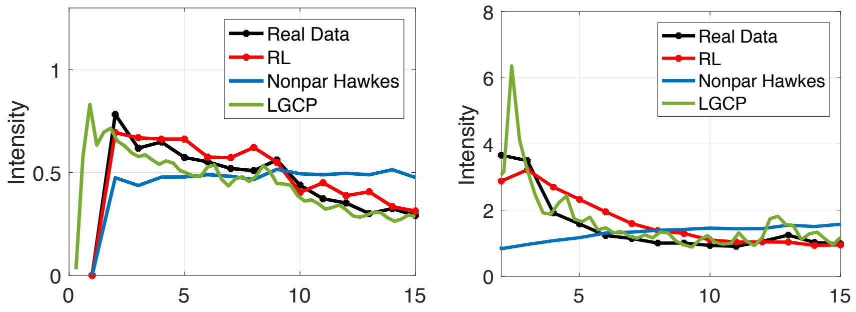
Figure 4.7: KS test results: CDF of p-values.

4.6.7 Comparison of Runtime.

The runtime for all methods averaged on all datasets is shown in Table 4.1. We note that both RLPP and WGANTPP are written in Tensorflow. However, WGANTPP adopts the adversarial training framework based on Wasserstein divergence, where both the generator and the discriminator are modeled as LSTMS. In contrast, RLPP only models the policy as a single LSTM with the reward function learned in an analytical form. As a consequence, RLPP requires less parameters and is more simpler to train while at the same time achieving comparable or even better performance.

Table 4.1: Comparison of runtime.

Method	**RLPP**	WGANTPP	RMTPP	SE	SC	IP
Time	80m	1560m	60m	2m	2m	2m
Ratio	40x	780x	30x	1x	1x	1x



(a): 911 dataset

(b): MIMIC dataset

Figure 4.8: Comparison of empirical intensity functions.

4.6.8 Comparisons to LGCP and non-parametric Hawkes.

We also compared RLPP to log-Gaussian Cox process (LGCP) model and non-parametric Hawkes with non-stationary background rate (Nonpar Hawkes) model regarding learned empirical intensity function. Representative comparison results are showed in Fig. 4.8. Our proposed method (RL) performs similarly to LGCP and outperforms Nonpar Hawkes on real datasets. However, LGCP needs to discretize time into windows and aggregate event into counts. This leads to some information loss and introduces additional tuning parameters. Moreover, the standard LGCP is not scalable, typically requiring $\mathcal{O}(n^3)$ in computation and $\mathcal{O}(n^2)$ in storage ($n =$ sequence # × window #). We used an implementation in GPy package[4], which requires 50% more time than our method (127 mins *vs* 80 mins) in processing 5% of the dataset. The nonparametric Hawkes model is parametrized by weighted sum of basis functions, similar to that of the inhomogeneous Poisson process baseline, and it is difficult to generalize outside the observation window.

[4] https://github.com/SheffieldML/GPy

4.7 Discussions

1. RMTPP we compared in experiments is a state-of-the-art maximum-likelihood-based model, which uses a similar RNN outputting parametrization of exponential distributions but fits the model parameters with maximum likelihood. Across our experiments over eight synthetic and real-world datasets, our proposed method performs consistently better than the MLE.

2. In theory, although MLE has many attractive limiting properties, it has no optimum properties for finite samples, in the sense that (when evaluated on finite samples) other estimators may provide a better estimate for the true parameters, e.g. [106]. Likelihood is related to KL divergence. Since KL divergence is asymmetric and has a number of drawbacks for finite sample (such as high variance and mode dropping), many other divergences have been proposed and shown to perform better in the finite sample case, e.g. [9]. Our proposed discrepancy is inspired by a similar use of RKHS discrepancy in two sample tests in [9]. RKHS discrepancy has been shown to perform nicely on finite sample and also preserve the asymptotic properties.

3. Another potential benefit of our proposed framework is that one may use the RNN to define a transformation for the temporal random variable instead of defining its output distribution. For example, we can establish our policy as a transformation of a sample from a unit rate exponential distribution. The same empirical objective in Eq. (4.8) will be used, but a different optimization algorithm is needed. Since no explicit parameterization of the output distribution is needed, this may lead to even more flexible models and this is left for future investigation.

4.8 Conclusions

This chaper proposes a reinforcement learning framework to learn point process models. We parametrized our policy as RNNs with stochastic neurons, which can sequentially sample discrete events. The policy is updated by directly minimizing the discrepancy between the generated sequences with the observed sequences, which can avoid model misspecification and the limitation of likelihood based approach. Furthermore, the discrepancy is explicitly evaluated in terms of the reward function in our setting. By choosing the function class of reward to be the unit ball in RKHS, we successfully derived an analytical optimal reward which maximizes the discrepancy. The optimal reward will iteratively encourage the policy to sample events as close as the observation. We show that our proposed approach performs well on both synthetic and real data.

CHAPTER 5
TEMPORAL LOGIC POINT PROCESSES

We propose a modeling framework for event data, which excels in small data regime with the ability to incorporate domain knowledge. Our framework will model the intensities of the event starts and ends via a set of first-order temporal logic rules. Using softened representation of temporal relations, and a weighted combination of logic rules, our framework can also deal with uncertainty in event data. Furthermore, many existing point process models can be interpreted as special cases of our framework given simple temporal logic rules. We derive a maximum likelihood estimation procedure for our model, and show that it can lead to accurate predictions when data are sparse and domain knowledge is critical.

5.1 Overview

A diverse range of application domains, such as healthcare [107], finance [108], smart city, and information networks [109, 83, 110], generate discrete events in continuous time. For instance, the occurrences of diseases on patients are event data; credit card uses are event data; the arrivals of passengers in subway systems are event data; and the posting and sharing of articles in online social platforms are also event data. Modeling these continuous-time event data becomes increasingly important to understand the underlying systems, to make an accurate prediction, and to regulate these systems towards desired states. Recently, sophisticated models such as recurrent Marked point processes [85], neural Hawkes processes [86] and reinforcement learning based methods [111] have been proposed, allowing us to model increasingly complex phenomena.

Although these models are very flexible, they require lots of data to properly fit the models, making these models perform poorly in the regime of small data. Furthermore, these models are notorious for their difficult-to-interpret predication results, and have been branded as “black boxes” [112]. This means it is difficult to clearly explain or identify the logic behind these predictions. In some cases, interpretability is more important than predictions. For example, in medicine, people are more interested in understanding what treatments contribute to the occurrences and cures of diseases than merely predicting the patients’ health status [113].

Very often, there already exists a rich collection of prior knowledge or logic rules from a particular domain, and we want to incorporate them to improve the interpretability and generalizability of the model. We want to fully utilize knowledge like this, rather than reinvent the wheel and purely relying on data to come up with the rule. Furthermore, when the amount of data is small and noisy, it will also be challenging to accurately recover these rules.

Thus our interest lies in interpretable event data modeling, and we want to incorporate prior temporal logic reasoning rules [114]. Our proposed modeling framework will explicitly model the durations of different types of events as random variables, and furthermore take into account the relations between different types of events specified by a set of temporal logic rules. More specifically, we will use two intensity functions to model the start and end of each type of event respectively, and these intensity functions are defined via a set of temporal logic rules involving both other types of events and temporal constraints.

In addition to the interpretability, our modeling framework has other characteristics and advantages:

(*i*) **Tolerance of uncertainty.** Data are noisy in the world, and time information is often imprecisely recorded. Treating logic rules as hard constraints will be too strict. Our model uses a weighted combination of logic rules, rather than using them as *hard constraints*. These designs allow us to deal with uncertainty and impreciseness of the rules for real-world data.

(*ii*) **Temporal relation constraints.** Our model can consider *temporal relation constraints* associated with logic rules, such as

- A happens *before* B.
- If A happens, and *after* 5 mins, B can happen.
- If A and B happen *simultaneously*, then *at the same time* C can happen.

Our model uses a softened parametrization of temporal relation constraints as part of our logic functions.

(*iii*) **Continuous-time reasoning process.** Our model captures the dynamics of a continuous-time reasoning process, and directly models the inter-arrival times of the evidence. Our model therefore can naturally deal with asynchronous events on the fly.

(*iv*) **Small data and knowledge transfer.** Our model better utilizes domain knowledge, and therefore will work on small datasets. Different datasets in similar concepts might share similar logic rules. We might leverage the learned logic weights in one dataset to warm-start the learning process on a different dataset. Our model makes it possible to transfer knowledge among different datasets.

Furthermore, we show that many existing point process models [115, 116, 117] can be recovered as special cases of our framework given simple temporal logic rules. We derive a maximum likelihood estimation procedure for our model, and show that it can lead to interpretable and accurate predictions in the regime of small data.

5.2 Temporal Logic

We first provides backgrounds for temporal logic reasoning.

5.2.1 First-order Logic

A **predicate** such as Smokes(c) or Friend(c, c') as a logic function $x(\cdot)$, is defined over a set of entities $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$, i.e., $x(\cdot) : \mathcal{C} \times \mathcal{C} \cdots \times \mathcal{C} \mapsto \{0, 1\}$. One can think of predicates as the *property* or *relation* of entities. A first-order **logic rule** is a logical connectives of predicates, such as

$$f_1 : \quad \forall c \;\; \text{Smokes}(c) \Rightarrow \text{Cancer}(c); \quad f_2 : \quad \forall c\, \forall c' \;\; \text{Friend(c, c')} \land \text{Smokes}(c) \Rightarrow \text{Smokes}(c').$$

Commonly used logical connectives are: $\land$ for conjunction, $\lor$ for disjunction, $\Rightarrow$ for implication, and $\neg$ for negation. Each first-order logic rule is also a logic function defined over the set of entities $\mathcal{C}$, i.e., $f(\cdot) : \; \mathcal{C} \times \mathcal{C} \cdots \times \mathcal{C} \mapsto \{0, 1\}$. For automated inference, it is often convenient to convert logic rules to a **clausal form**, which is a conjunction or disjunction of predicates. Table 5.1 demonstrates the fact that logic rule $x_A \Rightarrow x_B$ is logically equivalent to the clausal form $\neg x_A \lor x_B$. Every first-order logic rule can be converted to a clausal form using this mechanism. Generally, given predicates $x_{A_1}, \ldots, x_{A_k}, x_{B_1}, \ldots, x_{B_l}$, the first-order logic $(x_{A_1} \land x_{A_2} \cdots \land x_{A_k}) \Rightarrow (x_{B_1} \lor x_{B_2} \cdots \lor x_{B_l})$ is logically equivalent to $f : (\neg x_{A_1} \lor \neg x_{A_2} \cdots \lor \neg x_{A_k}) \lor (x_{B_1} \lor x_{B_2} \cdots \lor x_{B_l})$.

x_A, x_B	$x_A \Rightarrow x_B$	$\neg x_A \lor x_B$
0, 0	1	1
0, 1	1	1
1, 0	0	0
1, 1	1	1

Table 5.1: Logic rule in clausal form.

5.2.2 Temporal logic predicate

A **temporal predicate** is a logic function $x(\cdot, \cdot)$ over the set of entities $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ *and* time $t \in [0, \infty)$,

$$x(c, t) : \quad \mathcal{C} \times \mathcal{C} \cdots \times \mathcal{C} \times [0, \infty) \mapsto \{0, 1\},$$

which can only take two values, 0 or 1. For simplicity of notation, we will focus on the case with one entity, and drop the dependency of predicates on the entity. Hence, we will write $x(c, t)$ as $x(t)$ instead.

A temporal predicate $\{x(t)\}_{t\geq 0}$ can also be viewed as a **continuous-time two-state stochastic process**. For example, temporal predicate NormalBloodPressure(t) will take value 1 and 0 to indicate whether blood pressure is normal (0) or abnormal (1). The state transition time is stochastic.

Given a sample path of $\{x(t)\}_{t\geq 0}$ up to time t, the state transition time forms a partition of the time horizon. That is $\{x(t)\}_{t\geq 0}$ will stay in state 0 or state 1 for a **time interval**. For example, in Fig. 5.1 left, the grounded predicate is recorded as $x(t) = 0$ for $t \in [0, t_1)$, $x(t) = 1$ for $t \in [t_1, t_2)$, and so on. In some special cases, the grounded predicate $x(t)$ is instantaneous, we will obtain the point-based predicate process. Here, we regard point as a degenerate time interval. As in Fig 5.1 right, we record $x(t_1) = 1$, $x(t_2) = 1$, and so on at the jumping time. For other non-jumping time, $x(t) = 0$.

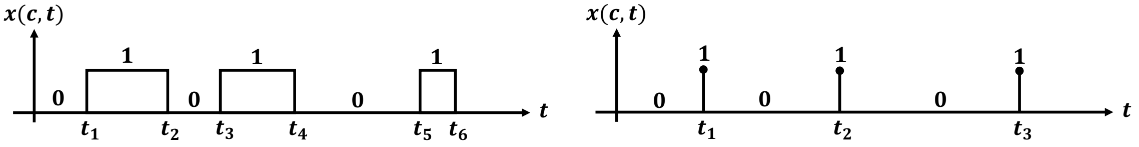


Figure 5.1: Left: Two-state temporal predicate. Right: Point-based temporal predicate.

Temporal relation

Allen's original paper [118] defined 13 types of **temporal relations between two time intervals**, $\{r_1, r_2, \ldots, r_{13}\}$, which are also mutually exclusive. More specifically, let two time intervals be $\tau_A = (t_{A_1}, t_{A_2}]$ and $\tau_B = (t_{B_1}, t_{B_2}]$ for predicate x_A and predicate x_B respectively, t_{A_1} and t_{B_1} be the respective interval starting times, and t_{A_2} and t_{B_2} be the respective interval ending times. Then a temporal relation is a logic function

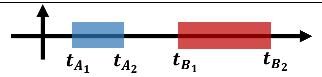
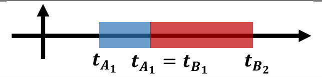
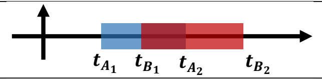
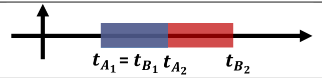
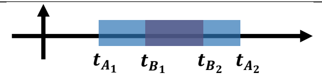
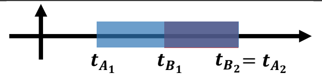
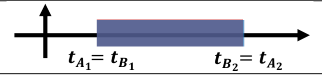
$$r(\cdot) : (t_{A_1}, t_{A_2}] \times (t_{B_1}, t_{B_2}] \mapsto \{0, 1\}$$

defined via

$$\text{a step function: } g(s) = \begin{cases} 1 & s \geq 0 \\ 0 & s < 0 \end{cases}, \quad \text{and an indicator function: } \kappa(s) = \begin{cases} 1 & s = 0 \\ 0 & o.w. \end{cases}, \tag{5.1}$$

for enforcing *hard* temporal constraints. Function forms of the 13 temporal relations can be founded in Table 5.2. Considering the inverses of relation $r_1 - r_6$ plus the symmetric relation r_7 "equal", there are a total of 13 relations. If there are *no* temporal relation constraints on A and B, then their temporal relations can take any of the 13 types, and $r_0 = r_{no}()$ returns the disjunction of these relations and is always "True" (i.e., 1).

Table 5.2: Interval-based temporal relation constraints and their illustrative figures.

Temporal Relation	Temporal Relations $r(\cdot)$	Illustration
$r_1 = r_{be}$: A before B	$g(t_{B_1} - t_{A_2})$	
$r_2 = r_{me}$: A meets B	$\kappa(t_{A_2} - t_{B_1})$	
$r_3 = r_{ov}$: A overlaps B	$g(t_{B_1} - t_{A_1}) \cdot g(t_{B_1} - t_{A_2}) \cdot g(t_{B_2} - t_{A_2})$	
$r_4 = r_{st}$: A starts B	$\kappa(t_{A_1} - t_{B_1}) \cdot g(t_{B_2} - t_{A_2})$	
$r_5 = r_{co}$: A contains B	$g(t_{B_1} - t_{A_1}) \cdot g(t_{A_2} - t_{B_2})$	
$r_6 = r_{fi}$: A finished-by B	$g(t_{B_1} - t_{A_1}) \cdot \kappa(t_{A_2} - t_{B_2})$	
$r_7 = r_{eq}$: A equals B	$\kappa(t_{A_1} - t_{B_1}) \cdot \kappa(t_{A_2} - t_{B_2})$	

More complex temporal relations can be decomposed as the composition of these 13 types of two way relations. For example, (A and B *before* C) can be decomposed as (A *before* C) and (B *before* C).

For degenerate point-based predicate process, where $t_{A_1} = t_{A_2} = t_A$, and $t_{B_1} = t_{B_2} =$

t_B, we will have a total of 3 types of temporal relations and their function forms, i.e.,

$$A \text{ before } B\text{: } g(t_B - t_A), \qquad A \text{ after } B\text{: } g(t_A - t_B), \qquad A \text{ equals } B\text{: } \kappa(t_A - t_B). \quad (5.2)$$

5.2.3 Temporal logic formula

Then a **temporal logic formula** is a logical composition of temporal logic predicates and temporal relations, $f(\mathcal{X}_f, \mathcal{T}_f) \mapsto \{0, 1\}$, where

- $\mathcal{X}_f = \{x_u(t)\}$ is a set of temporal predicates used to define the formula f,
- $\mathcal{T}_f = \{\tau_u\}$ is a set of time intervals, with each $x_u \in \mathcal{X}_f$ associated with a time interval $\tau_u = (t_{u_0}, t_{u_1}]$ (0 and 1 in the subscript indicates interval start and end respectively). We require that within time interval τ_u, the value of the temporal logic predicate $x_u(t)$ remains fixed.

Then a temporal logic formula have a generic form

$$f(\mathcal{X}, \mathcal{T}) = \left(\bigvee\nolimits_{x_u \in \mathcal{X}_f^+} x_u(t_{u_1})\right) \bigvee \left(\bigvee\nolimits_{v \in \mathcal{X}_f^-} \neg x_v(t_{v_1})\right) \bigwedge \left(\bigwedge\nolimits_{x_u, x_v \in \mathcal{X}_f} r_?(\tau_u, \tau_v)\right). \quad (5.3)$$

where $\mathcal{X}_f^-$ is the set of predicates used as negation in the formula f, $\mathcal{X}_f^+ = \mathcal{X} \setminus \mathcal{X}_f^-$, and $\{r_?(\tau_u, \tau_v)\}$ is a set of temporal relations between pairs of predicates. We use $r_?$ to indicate that the actual temporal relations used depend on specific formula.

5.3 Temporal Logic Point Processes

Suppose we have a collection of d temporal logic predicates $\boldsymbol{X} = \{x_1(t), x_2(t), \ldots, x_d(t)\}$, which is a compact representation of temporal knowledge base. An example of $\boldsymbol{X}$ in healthcare context is illustrated in Fig 5.2. Each predicate $x_u(t)$, defined as, UseDrug1(t), NormalBloodPressure(t) and so on, represent the properties, medical treatments, and health status of a patient at time $t > 0$.

The network topology of $\boldsymbol{X}$ is determined by a set of pre-defined temporal logic formulae $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$, which can express our *prior belief* on how these temporal predicates are related. For example in Fig 5.2, first-order logic rules such as "(NormalBloodPressure$(t) \wedge$ NormalHeartBeat$(t') \Rightarrow$ GoodSurvivalCondition$(t''))$ $\wedge\, r_{be}(t, t') \wedge r_{be}(t, t'')$" will define a clique in $\boldsymbol{X}$. We want to incorporate these temporal logic formulae in our point process model.

The advantages of our model are two-fold. First, the exact switching times of 0 or 1 for each process $\{x_u(t)\}_{t \geq 0}$ can be noisy or can contain uncertainty due to unmodeled effects. We are interested in modeling the statistical patterns of $\boldsymbol{X}$, and predict the values and the transition times of some temporal predicates in $\boldsymbol{X}$ (e.g., GoodSurvivalCondition). Second, each logic rules f_i is attached with a weight, indicating how confident is the rule in the world. The weights and temporal relation patterns of each logic rule will be learned from data.

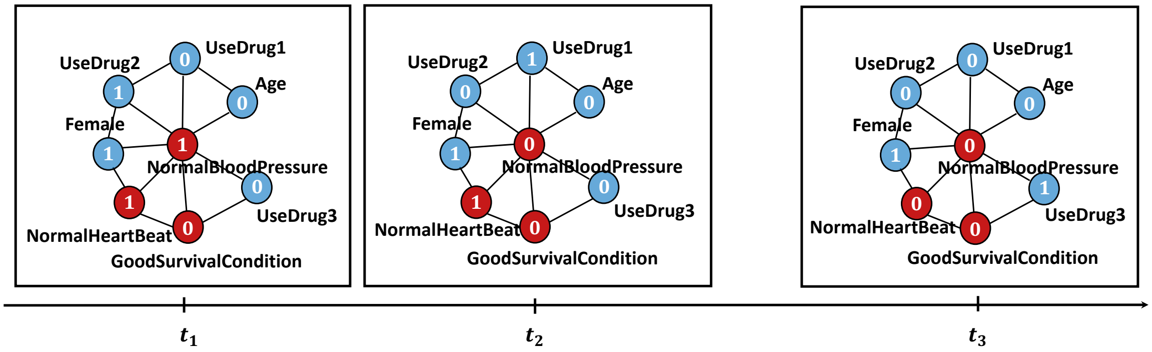


Figure 5.2: Illustration of temporal logic predicates $\boldsymbol{X}$ in medicine.

5.3.1 Dual intensity model for temporal predicate

We note that, for a temporal predicate, the positive and negative values will occur in an alternating fashion, dividing the time axis into segments. To facilitate later exposition, we will denote $\mathcal{H}_u(t)$ as the sequence of time intervals for each temporal predicate $x_u(t)$. More specifically, if we observe a sequence of transition time $\{t_1, t_2, \ldots, t_n\}$ between $(0, t]$, then we define

$$\mathcal{H}_u(t) := \{(0, t_1], (t_1, t_2], \ldots, (t_{n-1}, t_n], (t_n, t]\} \tag{5.4}$$

where values of the temporal predicate remain fixed with each time interval. One can also think of the length of each interval $t_{i+1} - t_i \geqslant 0$ is the dwell time of a particular fixed state.

Given the set of $\mathcal{H} = \{\mathcal{H}_u\}_{u=1,\ldots,d}$ for all temporal predicates, we can model the sequence of events for a particular temporal predicate using two intensity functions as illustrated in Fig. 5.3. More specially, define $\lambda_u^*(t) := \lambda(t|\mathcal{H}(t))$ the conditional transition intensity for "$x_u(t)$ transits from 0 to 1", and $\mu_u^*(t) := \mu(t|\mathcal{H}(t))$ the conditional transition intensity for "$x_u(t)$ transits from 1 to 0".

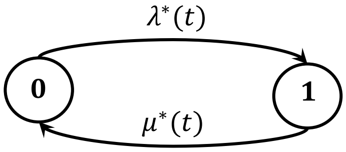


Figure 5.3: Two-state transition of temporal predicate.

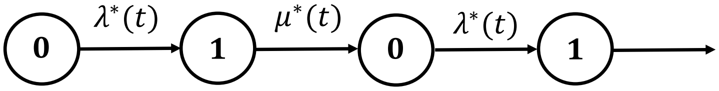


Figure 5.4: Unrolled chain: conditional process.

We can unroll the transition diagram and obtain a conditional process, with a unique sample path. All the transition intensities are time and history dependent. Suppose $x_u(t) = 0$ at $t = 0$, we will have the conditional process as displayed in Fig. 5.4.

5.3.2 Intensity guided by temporal logic rules

We will now discuss how to design the conditional transition intensity for temporal predicates by fusing a set of temporal logic formulae $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ from domain knowledge.

We will take a simple first-order temporal logic rule with temporal relation constraints as our running example. In plain language, a temporal reasoning rule for deducing event type C is

$$f_1: \quad (A \wedge B \Rightarrow C) \wedge (A \text{ before } B) \wedge (A \text{ and } B \text{ before } C), \tag{5.5}$$

which has the corresponding logical form as "if predicate x_A is true, and predicate x_B is true, then predicate x_C is true; furthermore, x_A occurs before x_B, and both occur before x_C". Write the temporal logic formula in clausal form as Eq. (5.3), we have

$$\begin{aligned} & f_1(x_A, x_B, x_C, t_A \in \tau_A, t_B \in \tau_B, t_C \in \tau_C): \\ := \ & (\neg x_A(t_{A_1}) \vee \neg x_B(t_{B_1}) \vee x_C(t_{C_1})) \wedge r_{be}(\tau_A, \tau_B) \wedge r_{be}(\tau_B, \tau_C) \end{aligned} \tag{5.6}$$

where we consider the value of predicate $x_A(t)$ in the time interval $\tau_A = (t_{A_1}, t_{A_2}]$, predicate $x_B(t)$ in the time interval $\tau_B = (t_{B_1}, t_{B_2}]$, and predicate $x_C(t)$ in the time interval $\tau_C = (t_{C_1}, t_{C_2}]$. Within these time intervals, predicates $x_A(t)$, $x_B(t)$ and $x_C(t)$ all maintain fixed values which may be different from each other.

We are interested in forward reasoning where we model the conditional transition intensity of deduced predicate x_C and treat the histories of x_A and x_B as evidence. For predicate $x_C(t)$, at any time t, it has two potential outcomes 0 or 1. One can observe only one, but not both, of the two potential outcomes. The unobserved outcome is called the "**counterfactual**" outcome. Suppose $x_C(t)$ is the "*observed*" outcome at time t, then $1 - x_C(t)$ is always the "*counterfactual*" outcome at time t.

To incorporate the knowledge from formula f_1 in constructing the transition intensity for x_C at time $t > 0$, we define a **formula effect** (FE) term as

$$\begin{aligned} \text{FE} = \delta_{f_1}(t \mid t_A \in \tau_A, t_B \in \tau_B) \;:=\; & f_1(x_A, x_B, 1 - x_C, t_A \in \tau_A, t_B \in \tau_B, t_C = t) \quad (5.7) \\ - \; & f_1(x_A, x_B, x_C, t_A \in \tau_A, t_B \in \tau_B, t_C = t) \end{aligned}$$

FE answers the question "what would happen if x_C transits its state given logic formula f_1 which takes into account the combination of historical states of other involved predicates". Note that the sign of FE can be 1, -1 or 0, which can be interpreted as

$$\text{sgn(FE)} = \begin{cases} 1 & \text{Positive effect to transit,} \\ -1 & \text{Negative effect to transit,} \\ 0 & \text{No effect to transit.} \end{cases}$$

In our example, we can check from the logic function that

$$\text{FE} = \begin{cases} (0,1] & \text{If observed } x_C(t) = 0,\ (x_A(t_{A_1}) = 1, x_B(t_{B_1}) = 1),\ (t_{A_2} < t_{B_1}),\ \text{and } (t_{B_2} < t) \\ [-1,0) & \text{If observed } x_C(t) = 1,\ (x_A(t_{A_1}) = 1, x_B(t_{B_1}) = 1),\ (t_{A_2} < t_{B_1}),\ \text{and } (t_{B_2} < t) \\ 0 & \text{Other combinations and temporal relations of } A \text{ and } B \end{cases}$$

Thus the conditional transition intensity for x_C from state 0 to 1, contributed by logic formula f_1 is

$$\lambda_C^*(t) = \exp\{w_{f_1} \cdot \underbrace{\sum_{\tau_A \in \mathcal{H}_A(t)} \sum_{\tau_B \in \mathcal{H}_B(t)} \delta_{f_1}(t \mid t_A \in \tau_A, t_B \in \tau_B)\}}_{\text{feature } \phi_{f_1}(t)}, \quad (5.8)$$

where the sign of the formula effect $\delta_{f_1}(t \mid t_A, t_B)$ indicates whether logic f_1 exerts a positive or negative effect provided the history $\mathcal{H}_A(t)$ and $\mathcal{H}_B(t)$, and the magnitude of $\delta_{f_1}(t_A, t_B, t)$ quantifies the strength of the influence. The double summation takes into account all combinations of temporal intervals in $\mathcal{H}_A(t)$ and $\mathcal{H}_B(t)$. One can think of the formula weight w_{f_1} as the confidence level on the formula. The higher the weight, the more influence the formula has on the intensity of $\lambda_C^*(t)$.

For conditional transition intensity $\mu_C^*(t)$, it has the same expression as Eq. (5.8). The only difference is that when we compute $\lambda_C^*(t)$, we let $x_C(t) = 0$, whereas when we compute $\mu_C^*(t)$, we let $x_C(t) = 1$, and this will yield different features. As illustrated in Fig. 5.5, the total valid (nonzero in terms of FE) combinations is 2, corresponding to $x_A(t_A) = 1$, $x_B(t_B) = 1$, and $x_A(t_A)$ happens before $x_B(t_B)$ and both before t. The feature can be evaluated from grounding $\delta_{f_1}(t|t_A, t_B)$ using Eq. (5.6) and (5.7).

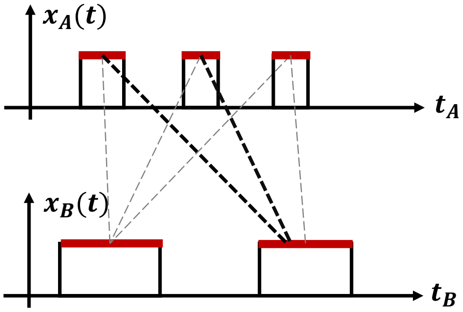


Figure 5.5: Effective combinations of A and B

Predicate x_C can be deduced from more than one logic formulae. For example, as shown in Fig. 5.6, x_C belongs to f_1 and f_2. We assume effect of temporal logic formula f_1 and f_2 are *additive* in designing the transition intensity for x_C.

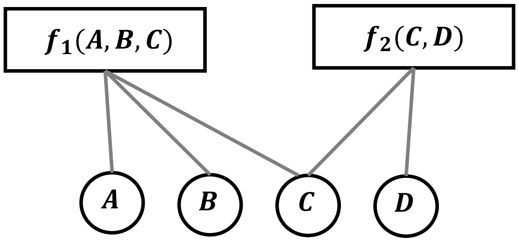


Figure 5.6: Factor graph

In general, given a set of temporal logic formulae $\mathcal{F}_C = \{f_1, \ldots, f_n\}$ for deducing

$x_C(t)$, the conditional transition intensity for predicate x_C, is designed as

$$\lambda_C^*(t) = \exp\left\{\sum_{f \in \mathcal{F}_C} w_f \cdot \phi_f(t) + b(t)\right\}, \tag{5.9}$$

where we also introduce a base temporal function $b(t)$ to always allow for spontaneous transition without influence from the logic. For instance, $b(t)$ can either be a constant $b(t) = b$, or a deterministic function of t. The expression of $\mu_C^*(t)$ is similar to Eq. (5.9), but with different values of features.

5.3.3 Softened temporal constraints

In practice, the temporal information usually cannot be accurately recorded in real time. It makes more sense to introduce **soft constraints** for the temporal relations. We introduce softened approximation functions for step function $g(s)$ and delta function $\kappa(s)$ in replacement of those used in the definitions of temporal relations in Table 5.2.

Step function $g(s)$ can be soften as a triangular function with area one or a logistic function,

$$g^{(soft)}(s) = \min(1, \max(0, \beta s + \tfrac{1}{2})), \text{ or } g^{(soft)}(s) = \frac{1}{1 + \exp(-\beta s)}. \tag{5.10}$$

Delta function $\kappa(s)$ can be soften as a triangular function with area one, or a Laplace distribution,

$$\kappa^{(soft)}(s) = \max(0, \min(\tfrac{s}{\gamma^2} + \tfrac{1}{\gamma}, -\tfrac{s}{\gamma^2} + \tfrac{1}{\gamma})), \text{ or } \kappa^{(soft)}(s) = \frac{\exp(-|s|/\gamma)}{\gamma}. \tag{5.11}$$

Parameters β and $\gamma \geq 1$ can be treated as unknown parameters, which can be learned from data.

5.3.4 Likelihood

By the definition of transition intensity in Eq. (5.9), we can write out the likelihood. For predicate C, given a realization of the process up to time t, as in Fig. 5.4, the likelihood $\mathcal{L}(\{x_C(t)\}_{t\geq 0})$ is

$$\lambda_C^*(t_1)\exp\left(-\int_0^{t_1}\lambda_C^*(s)ds\right)\cdot\mu_C^*(t_2)\exp\left(-\int_{t_1}^{t_2}\mu_C^*(s)ds\right)\cdots\exp\left(-\int_{t_n}^{t}\mu_C^*(s)ds\right), \tag{5.12}$$

provided predicate x_C starts in state 0 and stays in state 1 up to time t.

Sketch of proof. Let $p(t_{n+1}|\mathcal{H}_{t_n}, x_C(t_n) = 0)$ and $p(t_{n+1}|\mathcal{H}_{t_n}, x_C(t_n) = 1)$ be the conditional density function of the time of the next event t_{n+1} given the history of previous events $(t_0, t_1, \cdots, t_n)$ while $x_C(t_n) = 0$, and $x_C(t_n) = 1$ respectively. Let $F(t|\mathcal{H}_{t_n}, x_C(t_n) = 0)$, and $F(t|\mathcal{H}_{t_n}, x_C(t_n) = 1)$ be the corresponding cumulative distribution function for any $t > t_n$.

Based on the definition of the conditional transition intensity, we have

$$\lambda_C^*(t) = \frac{p(t|\mathcal{H}_{t_n}, x_C(t_n) = 0)}{1 - F(t|\mathcal{H}_t, x_C(t_n) = 0)}, \text{ and } \mu_C^*(t) = \frac{p(t|\mathcal{H}_{t_n}, x_C(t_n) = 1)}{1 - F(t|\mathcal{H}_{t_n}, x_C(t_n) = 1)} \tag{5.13}$$

From (5.13), we have

$$\lambda_C^*(t) = -\frac{d}{dt}\log(1 - F(t|\mathcal{H}_t, x_C(t_n) = 0)), \mu_C^*(t) = -\frac{d}{dt}\log(1 - F(t|\mathcal{H}_t, x_C(t_n) = 1)).$$

Integrating both sides, we can get

$$
\begin{aligned}
p(t_{n+1}|\mathcal{H}_{t_n}, x_C(t_n) = 0) &= \lambda_C^*(t) \exp\left(-\int_{t_n}^{t} \lambda_C^*(s)ds\right), \\
F(t|\mathcal{H}_{t_n}, x_C(t_n) = 0) &= 1 - \exp\left(-\int_{t_n}^{t} \lambda_C^*(s)ds\right), \\
p(t_{n+1}|\mathcal{H}_{t_n}, x_C(t_n) = 1) &= \mu_C^*(t) \exp\left(-\int_{t_n}^{t} \mu_C^*(s)ds\right), \\
F(t|\mathcal{H}_{t_n}, x_C(t_n) = 1) &= 1 - \exp\left(-\int_{t_n}^{t} \mu_C^*(s)ds\right).
\end{aligned}
$$

Let $t_0 = 0$. Given the initial state $x_C(t_0) = 0$,and the history of the trajectory $(t_1, t_2, \ldots, t_n)$, where $x_C(t_n) = 1$, the likelihood function can be factorized into all the conditional densities of each points given all points before it, i.e., $\mathcal{L}$ is

$$
\begin{aligned}
& p(t_1|\mathcal{H}_{t_0}, x_C(t_0) = 0) p(t_2|\mathcal{H}_{t_1}, x_C(t_1) = 1) \cdots \\
& \qquad p(t_{n-1}|\mathcal{H}_{t_{n-1}}, x_C(t_{n-1}) = 0)(1 - F(t|\mathcal{H}_{t_n}, x_C(t_n) = 1)) \\
= \ & \lambda_C^*(t_1) \exp\left(-\int_0^{t_1} \lambda_C^*(s)ds\right) \cdot \mu_C^*(t_2) \exp\left(-\int_{t_1}^{t_2} \mu_C^*(s)ds\right) \cdots \exp\left(-\int_{t_n}^{t} \mu_C^*(s)ds\right).
\end{aligned}
$$

Similarly, if let $t_0 = 0$, and given the initial state $x_C(t_0) = 0$,and $x_C(t_n) = 0$, the likelihood function $\mathcal{L}$ becomes

$$
\begin{aligned}
& p(t_1|\mathcal{H}_{t_0}, x_C(t_0) = 0) p(t_2|\mathcal{H}_{t_1}, x_C(t_1) = 1) \cdots \\
& \qquad p(t_{n-1}|\mathcal{H}_{t_{n-1}}, x_C(t_{n-1}) = 1)(1 - F(t|\mathcal{H}_{t_n}, x_C(t_n) = 0)) \\
= \ & \lambda_C^*(t_1) \exp\left(-\int_0^{t_1} \lambda_C^*(s)ds\right) \cdot \mu_C^*(t_2) \exp\left(-\int_{t_1}^{t_2} \mu_C^*(s)ds\right) \cdots \exp\left(-\int_{t_n}^{t} \lambda_C^*(s)ds\right),
\end{aligned}
$$

which completes the proof.

By considering all the predicates, the likelihood for the dataset is

$$\mathcal{L} \;=\; \prod_{u\in\{1,\ldots,d\}} \mathcal{L}(\{x_u(t)\}_{t\geq 0}).$$

All the unknown parameters regarding the logic weights $(\boldsymbol{w}_f, b)$ and the temporal relations β and γ will be jointly learned by maximizing the likelihood.

5.4 Experiments

We will demonstrate the **accuracy**, **flexibility** and **interpretability** of temporal logic point process models. We first show that we can use *simple logic rules* to recover several well-known parametric point processes. Then we use simple rules for a three-player game to generate complex temporal event patterns. Finally, we evaluate the interpretability of our model on a real healthcare dataset.

5.4.1 Recover temporal point processes

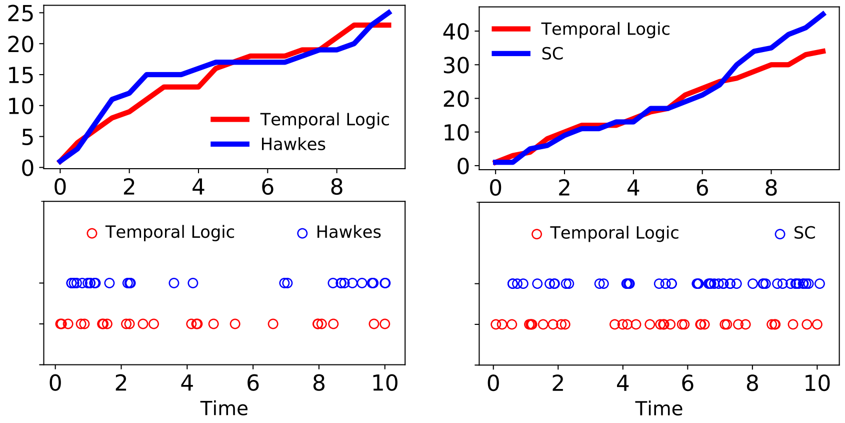
We show the flexibility and accuracy of our model by recovering nonlinear Hawkes processes and self-correcting processes from data. The training data is one sequence of events generated from nonlinear Hawkes process and self-correcting processes, respectively.

(*i*) **Hawkes.** The intensity function $\lambda(t) = b + \alpha \sum_{t_i<t} \exp(-(t-t_i))$, where $b > 0$ and $\alpha > 0$, means that previous events will boost the occurrence of new events. This will correspond to "If A happens, then A will happen again afterwards", which can be expressed as a first-order temporal logic rule

$$f_{\text{Hawkes}}(x_A(t), x_A(t'), t=t, t'=t') : \quad (\neg x_A(t) \vee x_A(t')) \wedge r_{be}(t,t'),$$

where $x_A(t)$ is a degenerate point-based temporal predicate. Furthermore, in the intensity,

α corresponds to formula weight, and we have used softened temporal relation by Logistic function.



(a): Hawkes. (b) Self-correcting.

Figure 5.7: Generated events v.s. training events.

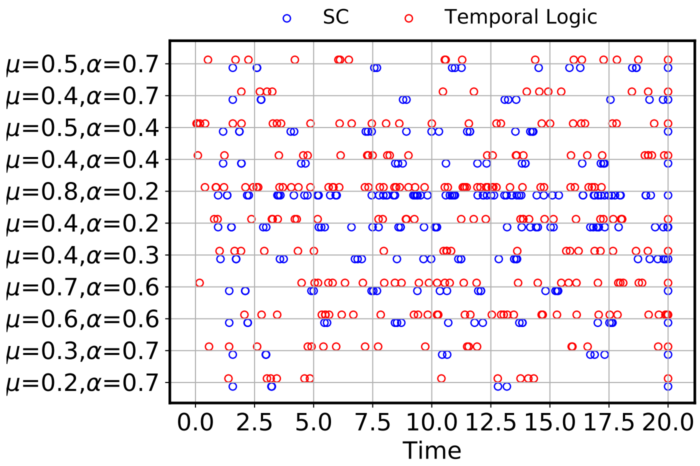


Figure 5.8: More self-correcting processes examples

(*ii*) **Self-correcting.** The intensity function $\lambda(t) = \exp\left(bt - \sum_{t_i<t} \alpha\right)$, where $\mu > 0$ and $\alpha > 0$ are positive parameters, models that previous events will inhibit the occurrence of new events. This will correspond to “If A happens, then A will not happen again”, which can be expressed as a first-order temporal logic rule

$$f_{\text{self-correcting}}(x_A(t), x_A(t'), t = t, t' = t') : \quad (\neg x_A(t) \vee \neg x_A(t')) \wedge r_{be}(t, t').$$

where $x_A(t)$ is a degenerate point-based temporal predicate. Furthermore, in the intensity, α corresponds to formula weight.

In our experiment, we use the above hypothesized temporal logic rules to design the intensity of temporal logic point processes. To verify the accuracy of these temporal logic rules and our model, we generate events from the learned model and compare the cumulative event counts to the training sequences. As displayed in Fig. 5.7 and Fig. 5.8, only using a very short sequence of events, our temporal logic point processes can accurately recover the dynamics of nonlinear Hawkes and self-correcting processes.

5.4.2 Three-player game.

We design a game, where player A, B and C follow the following logic rules,

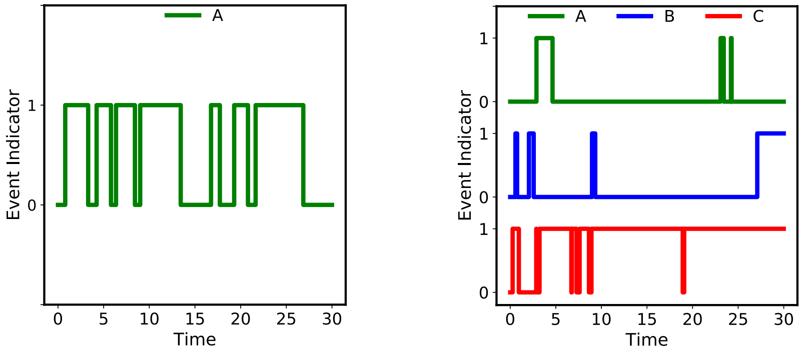
$$f_1: \ (\neg x_C(t) \wedge x_A(t) \Rightarrow x_A(t')) \wedge r_{be}(t, t'); \ f_2: \ x_A(t) \wedge x_B(t) \Rightarrow x_C(t); \ f_3: \ x_C(t) \Rightarrow \neg x_A(t).$$

For player A, if there is no C, it will occur periodically, which corresponds to a temporal logic rule

$$f_{\text{periodic}}: \qquad (\neg x_A(t) \vee x_A(t')) \wedge r_{\text{eq}}(t, t' - T),$$

where T is the period. We simulate this repeated game after player C joins the game. Furthermore, we will also use softened temporal relation to represent r_{eq} using Laplace kernel.

As illustrated in Fig. 5.9, once A and B occur, C will be triggered (f_2). However, C will inhibit A (f_3). Then A stops happening, and C vanishes as a result (f_2). After C disappears, A occurs again (f_1). This simple example shows the flexibility of our temporal logic model. The simulated dynamic systems, governed by logic rules, exhibit different stages automatically, and demonstrate the flexibility of the model.



(a): Periodic A without C in the game. (b) Involving C in the game.

Figure 5.9: Repeated three-player game.

5.4.3 Healthcare applications.

We demonstrate the interpretability and prediction accuracy of our temporal logic model on MIMIC-III dataset [119]. A total of 100 sepsis patients (with mean age 66.6, 43.6% female) are selected as our population.

To establish our model, 31 logic rules, as shown in Table 5.3, are introduced as prior knowledge. These logic rules are collected from real observed treatments as well as domain knowledge. Predicates $\{x_i\}_{i=1,\ldots,23}$ denote different types of treatments (i.e., drugs, and see Appendix for details), u_1 denotes the blood pressure, and u_2 is the survival condition. Defined by the temporal logic rules, "treatments", "blood pressure", and "survival condition" are inter-related and the transition intensity of these predicates can be constructed. All predicates take values 0 or 1. For drugs, 1 means the treatment is applied, and 0 otherwise; for blood pressures, 1 means normal status and 0 otherwise; for survival condition, 1 indicates survival and 0 otherwise. All predicates will be grounded sequentially with state transition times recorded.

To evaluate our model's prediction accuracy on small data, we train our model using only 5 and 30 patients' trajectories respectively, and predict the real-time states of u_1 and u_2 on test patients. We make a comparison with LSTM and RNN, which are state-of-the-art predictive models, and the results are summarized in Table 5.4. Our model performs fairly well and consistently better than the baselines, due to better utilization of prior knowledge.

$f: \ (x_i \Rightarrow u_2) \wedge \ r_{\mathfrak{b}}(t_{x_i}, t_{u_2}) \ , i \in \{1, 2, ..., 23\}$
$f: \ (x_j \Rightarrow u_1) \wedge \ r_{\mathfrak{b}}(t_{x_j}, t_{u_1}), j \in \{2, 11, 12, 19\}$
$f: \ (u_1 \Rightarrow u_2) \wedge \ (t_{u_1} = t_{u_2})$
$f: \ (x_{10} \wedge x_{20} \Rightarrow u_2) \wedge \ r_{\mathfrak{b}}(t_{x_{10}}, t_{u_2}) \wedge \ r_{\mathfrak{b}}(t_{x_{20}}, t_{u_2})$
$f: \ (x_{14} \wedge x_{20} \Rightarrow u_2) \ \wedge \ r_{\mathfrak{b}}(t_{x_{14}}, t_{u_2}) \wedge \ r_{\mathfrak{b}}(t_{x_{20}}, t_{u_2})$
$f: \ (\neg x_{12} \wedge x_8 \Rightarrow u_2) \wedge \ r_{\mathfrak{b}}(t_{x_{12}}, t_{u_2}) \wedge \ r_{\mathfrak{b}}(t_{x_8}, t_{u_2})$
$f: \ (\neg x_{12} \wedge x_{17} \Rightarrow u_2) \ \wedge \ r_{\mathfrak{b}}(t_{x_{12}}, t_{u_2}) \wedge \ r_{\mathfrak{b}}(t_{x_{17}}, t_{u_2})$

Table 5.3: List of logic rules.

Table 5.4: BP and mortality prediction

Method	Train/Test: 5/5		Train/Test: 30/10	
	BP Precision	Mortality Precision	BP Precision	Mortality Precision
LSTM	0.264±0.036	0.505±0.371	0.242±0.034	0.545±0.325
RNN	0.217±0.057	0.517±0.097	0.213±0.035	0.557±0.245
Temporal Logic	**0.535±0.012**	**0.641±0.037**	**0.599±0.014**	**0.658±0.019**

We are also interested in understanding what types of medical treatments contribute more to the outcome. The learned formula weights based on the population are reported in Appendix.

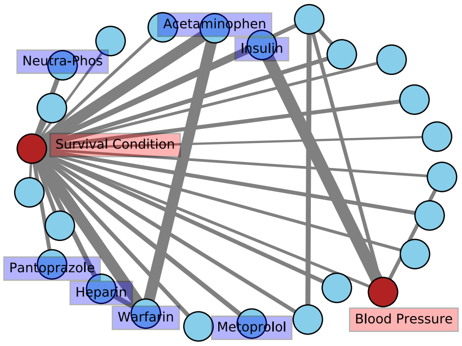


Figure 5.10: Formula graph.

In Fig. 5.10, each node represents a predicate and the thickness of the lines represent the weights of the formula. We labeled the Blood pressure and Survival condition predicates and discovered important drugs. We discovered the rule $f: \ x_{11} \Rightarrow u_1$, where x_{11} is insulin, is the most important factor to affect blood pressure. Insulin therapy has been verified that may increase blood pressure levels [120]. This discovery is consistent with domain knowledge that the physiologically frail diabetic individuals suffer the highest infection rates of sepsis [121]. Another example is $f: \ x_{10} \Rightarrow x_{20}$ where x_{10} is Acetaminophen and x_{20} is Warfarin. Warfarin is a vitamin K antagonist and Acetaminophen has been shown that may enhance the anticoagulant effect of Warfarin [122]. In Fig, 5.11, we visualized how the logic weights are updated in the training process. At the beginning, the logic weights are almost the same. With more iterations, the dominant rules appeared. These results show that our model can better predict individual patient's health status, and can uncover important rules using population data.

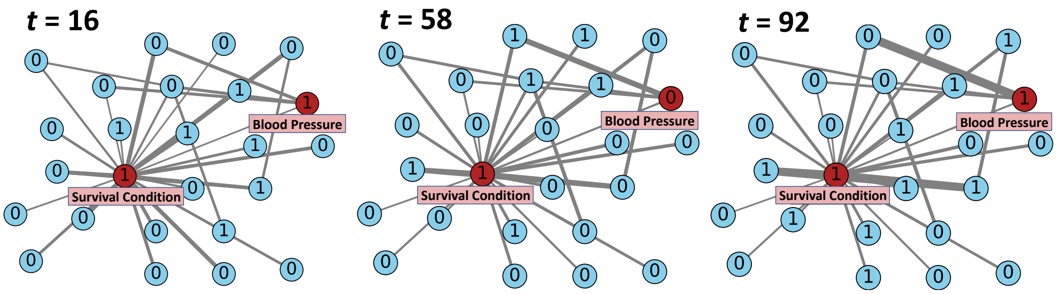


Figure 5.11: Weights during training

5.5 Discussion.

In this chapter, we proposed a unified framework to integrate first-order temporal logic rules into point processes. Our model is easy to interpret and works well on small data. We also introduced a softened representation of the temporal relation constraints to tolerate uncertainty. Many existing point processes can be recovered by defining simple logic rules. As for future work, we aim to introduce latent predicates to make our model more flexible.

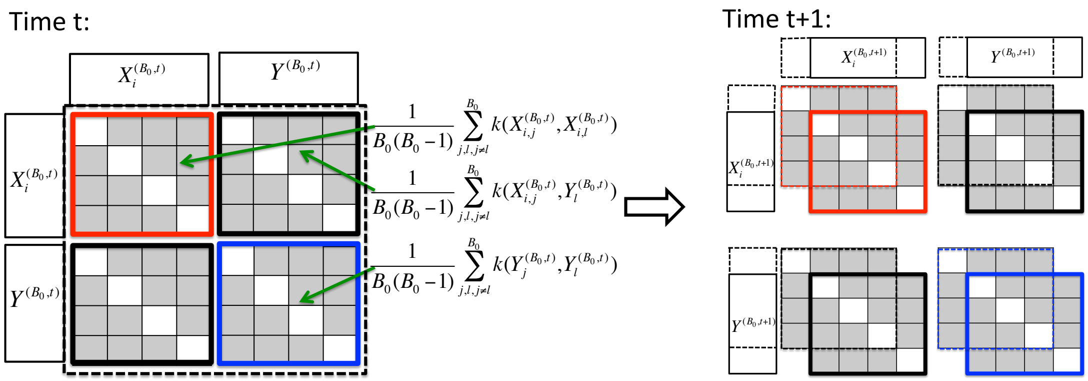
Appendices

APPENDIX A
SCAN B-STATISTIC FOR KERNEL CHANGE-POINT DETECTION

A.1 Recursive implementation of online scan B-statistic

The online scan B-statistic can be computed recursively via a simple update scheme. By its construction, when time elapses from t to $(t+1)$, a new sample is added into the post-change block, and the oldest sample is moved to the reference pool. Each reference block is updated similarly by adding one sample randomly drawn from the pool of reference data, and the oldest sample is purged. Hence, only a limited number of entries in the Gram matrix due to the new sample will be updated.

The update scheme is illustrated in Fig. A.1 and explained in more details therein. The online B-statistic is formed with N background blocks and one testing block and, hence, we keep track of N Gram matrices. For illustration purposes, we partition the Gram matrix into four windows (in red, black and blue, as shown on the left panel). At time t, to obtain $\mathrm{MMD}^2(X_i^{(B_0,t)}, Y^{(B_0,t)})$, we compute the shaded elements and take an average within each window. The diagonal entries in each window are removed to obtain an unbiased estimate. At time $t+1$, we update $X_i^{(B_0,t)}$ and $Y^{(B_0,t)}$ with the new data point and purge the oldest data point, and update the Gram matrix by moving the colored window as shown on the right panel. We compute the elements within the new windows, and take an average. Note that we only need to compute the right-most column and the bottom row. Similarly, the offline scan B-statistic can also be computed recursively by utilizing the fact that Z_B for $B \in \{2, \ldots, B_{\max}\}$ shares many common terms.



Time t:
$X_i^{(B_0,t)}$
$Y^{(B_0,t)}$
$X_i^{(B_0,t)}$
$Y^{(B_0,t)}$
$\frac{1}{B_0(B_0-1)}\sum_{j,l,j\neq l}^{B_0} k(X_{i,j}^{(B_0,t)}, X_{i,l}^{(B_0,t)})$
$\frac{1}{B_0(B_0-1)}\sum_{j,l,j\neq l}^{B_0} k(X_{i,j}^{(B_0,t)}, Y_l^{(B_0,t)})$
$\frac{1}{B_0(B_0-1)}\sum_{j,l,j\neq l}^{B_0} k(Y_j^{(B_0,t)}, Y_l^{(B_0,t)})$
Time t+1:
$X_i^{(B_0,t+1)}$
$Y^{(B_0,t+1)}$
$X_i^{(B_0,t+1)}$
$Y^{(B_0,t+1)}$

Figure A.1: Recursive update scheme to compute the online scan B-statistics.

A.2 Variance and covariance calculation

Below, $X_{i,j}^{(B)}$, where $i = 1, \ldots, N$, and $j = 2, \ldots, B_{\max}$, denotes the j-th sample in the i-th block $X_i^{(B)}$, and $Y_j^{(B)}$ denotes the j-th sample in $Y^{(B)}$. The superscript B denotes the block size. We start with proving Lemma A.2.1 and Lemma A.2.2, which are useful in proving Lemma 1.

Lemma A.2.1 (Variance of MMD, under the null.) *Under the null hypothesis,*

$$\mathrm{Var}\left[\mathrm{MMD}^2(X_i^{(B)}, Y^{(B)})\right] = \binom{B}{2}^{-1} \mathbb{E}[h^2(x, x', y, y')], \quad i = 1, \ldots, N. \tag{A.1}$$

Proof For notational simplicity, below we drop the superscript B, which denotes the block size. Furthermore, we use x, x', y and y' to denote generic samples, i.e., $X_{i,l} \overset{d}{=} x$, $X_{i,j} \overset{d}{=} x'$, $Y_l \overset{d}{=} y$, $Y_j \overset{d}{=} y'$ and they are mutually independent of each other. Here the notation $\overset{d}{=}$ means two random variables have the same distribution. Below, we follow the same convention. For any $i = 1, 2, \ldots, n$, by definition of U-statistic, we have

$$\begin{aligned}\text{Var}\left[\text{MMD}^2(X_i, Y)\right] &= \text{Var}\left[\binom{B}{2}^{-1} \sum_{l<j} h(X_{i,l}, X_{i,j}, Y_l, Y_j)\right] \\ &= \binom{B}{2}^{-2} \left[\binom{B}{2}\binom{2}{1}\binom{B-2}{2-1}\text{Var}\left[\mathbb{E}_{x,y}[h(x, x', y, y')]\right]\right. \\ &\quad \left.+\binom{B}{2}\binom{2}{2}\binom{B-2}{2-2}\text{Var}\left[h(x, x', y, y')\right]\right].\end{aligned} \tag{A.2}$$

Under null distribution, $\mathbb{E}_{x,y}[h(x, x', y, y')] = 0$. Thus, $\text{Var}\left[\mathbb{E}_{x_iy}[h(x, x', y, y')]\right] = 0$, and

$$\text{Var}\left[h(x, x', y, y')\right] = \mathbb{E}[h^2(x, x', y, y')] - \mathbb{E}[h(x, x', y, y')]^2 = \mathbb{E}[h^2(x, x', y, y')].$$

Substitute these results into (A.2), and we obtain the desired result (A.1).

■

Lemma A.2.2 (Covariance of MMD, under the null, different block index.) *For $s \neq 0$, under null hypothesis*

$$\begin{aligned}&\text{Cov}\left[\text{MMD}^2(X_i^{(B)}, Y^{(B)}), \text{MMD}^2(X_{i+s}^{(B)}, Y^{(B)})\right] \\ =&\binom{B}{2}^{-1}\text{Cov}\left[h(x, x', y, y'), h(x'', x''', y, y')\right].\end{aligned}$$

Proof For $i = 1, 2, \ldots, N$, and $s = (1-i), (2-i), \ldots, (N-i), s \neq 0$,

$$\begin{aligned}&\text{Cov}\left[\text{MMD}^2(X_i, Y), \text{MMD}^2(X_{i+s}, Y)\right] \\ &= \text{Cov}\left[\binom{B}{2}^{-1} \sum_{l<j} h(X_{i,l}, X_{i,j}, Y_l, Y_j), \binom{B}{2}^{-1} \sum_{p<q} h(X_{i+s,p}, X_{i+s,q}, Y_p, Y_q)\right] \\ &= \binom{B}{2}^{-2}\binom{B}{2}\binom{2}{1}\binom{B-2}{2-1}\text{Cov}\left[h(x, x', y, y'), h(x'', x''', y, y'')\right] \\ &\quad + \binom{B}{2}^{-2}\binom{B}{2}\binom{2}{2}\binom{B-2}{2-2}\text{Cov}\left[h(x, x', y, y'), h(x'', x''', y, y')\right].\end{aligned}$$

Under null distribution,

$$
\begin{aligned}
& \operatorname{Cov}\left[h(x,x',y,y'),h(x'',x''',y,y'')\right] \\
= & \int h(x,x',y,y')h(x'',x''',y,y'')d\mathbb{P}(x,x',x'',x''',y,y',y'') \\
= & \int \left(\underbrace{\int h(x,x',y,y')d\mathbb{P}(x',y')}_{=0} \right) d\mathbb{P}(x) \cdot \int \left(\underbrace{\int h(x'',x''',y,y'')d\mathbb{P}(x'',y'')}_{=0} \right) d\mathbb{P}(x''') = 0.
\end{aligned}
$$

Above, with a slight abuse of notation, we use $d\mathbb{P}(\cdot)$ to denote the probability measure of appropriate arguments. Finally, we have the desired results as shown in Lemma A.2.2.

■

A.2.1 Variance of scan B-statistics

Proof [Proof for Lemma 1] Using results in Lemma A.2.1 and Lemma A.2.2, we have

$$
\begin{aligned}
\operatorname{Var}[Z_B] &= \operatorname{Var}\left[\frac{1}{N}\sum_{i=1}^{N}\operatorname{MMD}^2(X_i,Y)\right] \\
&= \frac{1}{N^2}\left[N\operatorname{Var}[\operatorname{MMD}^2(X_i,Y)] + \sum_{i\neq j}\operatorname{Cov}\left[\operatorname{MMD}^2(X_i,Y;B),\operatorname{MMD}^2(X_j,Y)\right]\right] \\
&= \binom{B}{2}^{-1}\left[\frac{1}{N}\mathbb{E}[h^2(x,x',y,y')] + \frac{N-1}{N}\operatorname{Cov}\left[h(x,x',y,y'),h(x'',x''',y,y')\right]\right].
\end{aligned}
$$

■

Next, we introduce Lemma A.2.3 and Lemma A.2.4, which are useful in proving Lemma 3.

Lemma A.2.3 (Covariance of MMD, different block sizes, same block index.) *For blocks* *with the same index i but with distinct block sizes, under the null hypothesis we have*

$$\mathrm{Cov}\left[\mathrm{MMD}^2(X_i^{(B)}, Y^{(B)}), \mathrm{MMD}^2(X_i^{(B+v)}, Y^{(B+v)})\right] = \binom{B \vee (B+v)}{2}^{-1} \mathbb{E}[h^2(x, x', y, y')]. \tag{A.3}$$

Proof Note that

$$\begin{aligned}
& \mathrm{Cov}\left[\mathrm{MMD}^2(X_i^{(B)}, Y^{(B)}), \mathrm{MMD}^2(X_i^{(B+v)}, Y^{(B+v)})\right] \\
= \ & \mathrm{Cov}\left[\binom{B}{2}^{-1} \sum_{l<j}^{B} h(X_{i,l}, X_{i,j}, Y_l, Y_j), \binom{B+v}{2}^{-1} \sum_{p<q}^{B+v} h(X_{i,p}, X_{i,q}, Y_p, Y_q)\right] \\
= \ & \binom{B}{2}^{-1} \binom{B+v}{2}^{-1} \mathrm{Cov}\left[\sum_{l<j}^{B} h(X_{i,l}, X_{i,j}, Y_l, Y_j), \sum_{p<q}^{B+v} h(X_{i,p}, X_{i,q}, Y_p, Y_q)\right] \\
= \ & \binom{B}{2}^{-1} \binom{B+v}{2}^{-1} \binom{B \wedge (B+v)}{2} \mathrm{Var}[h(x, x', y, y')] \\
= \ & \binom{B \vee (B+v)}{2}^{-1} \mathbb{E}[h^2(x, x', y, y')],
\end{aligned}$$

where the second last equality is due to a similar argument as before to drop block indices as they are *i.i.d.* under the null. ■

Lemma A.2.4 (Covariance of MMD, different block sizes, different block indices.) *Under the null we have*

$$\mathrm{Cov}\left[\mathrm{MMD}^2(X_i^{(B)}, Y^{(B)}), \mathrm{MMD}^2(X_{i+s}^{(B+v)}, Y^{(B+v)})\right] = \binom{B \vee (B+v)}{2}^{-1} \cdot \\ \mathrm{Cov}\left[h(x, x', y, y'), h(x'', x''', y, y')\right].$$

Proof Note that

$$
\begin{aligned}
&\mathrm{Cov}\left[\mathrm{MMD}^2(X_i^{(B)}, Y^{(B)}), \mathrm{MMD}^2(X_{i+s}^{(B+v)}, Y^{(B+v)})\right] \\
=\ & \mathrm{Cov}\left[\binom{B}{2}^{-1}\sum_{l<j}^{B} h(X_{i,l}^{(B)}, X_{i,j}^{(B)}, Y_l^{(B)}, Y_j^{(B)}), \binom{B+v}{2}^{-1}\sum_{p<q}^{B+v} h(X_{i+s,p}^{(B+v)}, X_{i+s,q}^{(B+v)}, Y_p^{(B+v)}, Y_q^{(B+v)})\right] \\
=\ & \binom{B}{2}^{-1}\binom{B+v}{2}^{-1}\mathrm{Cov}\left[\sum_{l<j}^{B} h(X_{i,l}^{(B)}, X_{i,j}^{(B)}, Y_l^{(B)}, Y_j^{(B)}), \sum_{p<q}^{B+v} h(X_{i+s,p}^{(B+v)}, X_{i+s,q}^{(B+v)}, Y_p^{(B+v)}, Y_q^{(B+v)})\right] \\
=\ & \binom{B}{2}^{-1}\binom{B+v}{2}^{-1}\binom{B\wedge(B+v)}{2}\mathrm{Cov}\left[h(x,x',y,y'), h(x'',x''',y,y')\right] \\
=\ & \binom{B\vee(B+v)}{2}^{-1}\mathrm{Cov}\left[h(x,x',y,y'), h(x'',x''',y,y')\right],
\end{aligned}
$$

where the second last equality is due to a similar argument as before to drop block indices as they are *i.i.d.* under the null.

■

A.2.2 Covariance of offline scan B-statistics.

Proof [Proof of Lemma 3] For the offline case, we have that the correlation

$$
r_{B,B+v} := \frac{1}{\sqrt{\mathrm{Var}[Z_B]}}\frac{1}{\sqrt{\mathrm{Var}[Z_{B+v}]}}\mathrm{Cov}\left[Z_B, Z_{B+v}\right],
$$

where

$$
\begin{aligned}
\mathrm{Cov}\left(Z_B, Z_{B+v}\right) &= \mathrm{Cov}\left[\frac{1}{N}\sum_{i=1}^{N}\mathrm{MMD}^2(X_i^{(B)}, Y^{(B)}), \frac{1}{N}\sum_{j=1}^{n}\mathrm{MMD}^2(X_j^{(B+v)}, Y^{(B+v)})\right] \\
&= \frac{1}{N}\mathrm{Cov}\left[\mathbf{MMD}^2(X_i^{(B)}, Y^{(B)}), \mathbf{MMD}^2(X_i^{(B+v)}, Y^{(B+v)})\right] \\
&\quad + \frac{1}{N^2}\sum_{i\neq j}\mathrm{Cov}\left[\mathbf{MMD}^2(X_i^{(B)}, Y^{(B)}), \mathbf{MMD}^2(X_j^{(B+v)}, Y^{(B+v)})\right].
\end{aligned}
$$

Using results from Lemma A.2.3 and Lemma A.2.4, we have:

$$\begin{aligned}\mathrm{Cov}\left(Z_B, Z_{B+v}\right) = \binom{B \vee (B+v)}{2}^{-1} & \left[\frac{1}{N}\mathbb{E}[h^2(x,x',y,y')]\right.\\ & \left.+\frac{N-1}{N}\mathrm{Cov}\left[h(x,x',y,y'), h(x'',x''',y,y')\right]\right].\end{aligned}$$

Finally, plugging in the expressions for $\mathrm{Var}[Z_B]$ and $\mathrm{Var}[Z_{B+v}]$, we have (2.10) for the offline case.

A.2.3 Covariance of online scan B-statistic

Similarly, for the online case we need to analyze $\rho_{t,t+s} := \mathrm{Cov}\left(Z'_{B_0,t}, Z'_{B_0,t+s}\right)$. We adopt the same strategy as the above for a fixed block size B_0 to obtain

$$\begin{aligned}& \mathrm{Cov}\left(\mathrm{MMD}^2(X_i^{(B_0,t)}, Y^{(B_0,t)}), \mathrm{MMD}^2(X_i^{(B_0,t+s)}, Y^{(B_0,t+s)})\right)\\ = \;& \mathrm{Cov}\left[\binom{B_0}{2}^{-1}\sum_{l<j}^{B_0} h(X_{i,l}^{(t)}, X_{i,j}^{(t)}, Y_l^{(t)}, Y_j^{(t)}), \binom{B_0}{2}^{-1}\sum_{p<q}^{B_0} h(X_{i,p}^{(t+s)}, X_{i,q}^{(t+s)}, Y_p^{(t+s)}, Y_q^{(t+s)})\right]\\ = \;& \binom{B_0}{2}^{-2}\binom{(B_0-s)\vee 0}{2}\mathrm{Var}[h(x,x',y,y')]. \qquad \text{(A.4)}\end{aligned}$$

Figure A.2 (a) demonstrates how $\mathrm{MMD}^2(X_i^{(B_0,t)}, Y^{(B_0,t)})$ and $\mathrm{MMD}^2(X_i^{(B_0,t+s)}, Y^{(B_0,t+s)})$ are constructed. The shaded areas represent the overlapping data.

Similarly, we have

$$\begin{aligned}& \mathrm{Cov}\left(\mathrm{MMD}^2(X_i^{(B_0,t)}, Y^{(B_0,t)}), \mathrm{MMD}^2(X_j^{(B_0,t+s)}, Y^{(B_0,t+s)})\right)\\ = \;& \mathrm{Cov}\left[\binom{B_0}{2}^{-1}\sum_{l<k}^{B_0} h(X_{i,l}^{(t)}, X_{i,k}^{(t)}, Y_l^{(t)}, Y_k^{(t)}), \binom{B_0}{2}^{-1}\sum_{p<q}^{B_0} h(X_{j,p}^{(t+s)}, X_{j,q}^{(t+s)}, Y_p^{(t+s)}, Y_q^{(t+s)})\right]\\ = \;& \binom{B_0}{2}^{-2}\binom{(B_0-s)\vee 0}{2}\mathrm{Cov}(h(x,x',y,y'), h(x'',x''',y,y')), \qquad \text{(A.5)}\end{aligned}$$

Figure A.2 (b) demonstrates how $\mathrm{MMD}^2(X_i^{(B_0,t)}, Y^{(B_0,t)})$ and $\mathrm{MMD}^2(X_j^{(B_0,t+s)}, Y^{(B_0,t+s)})$,

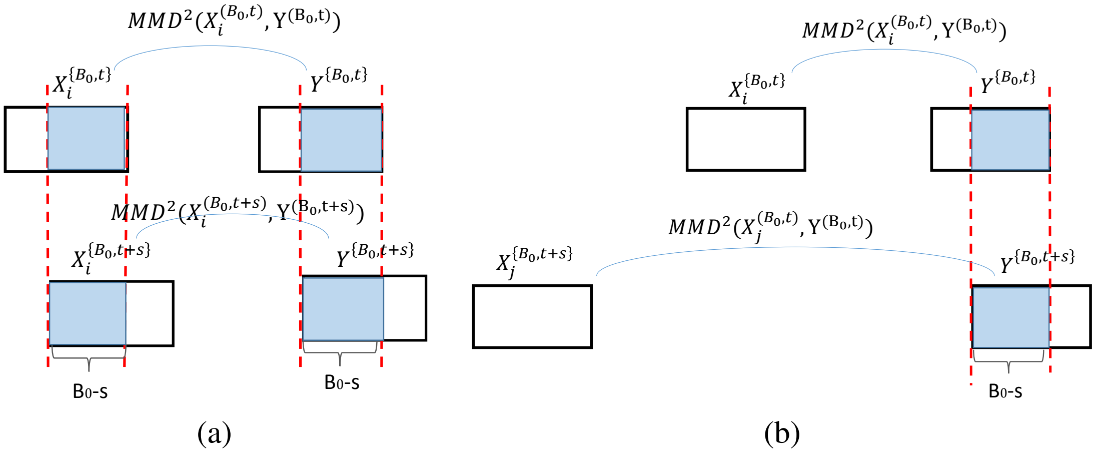


Figure A.2: Illustration of how MMD^{2}s are constructed.

$j \neq i$ are constructed. The shaded areas represent the overlapping data. Thus,

$$
\begin{aligned}
& \operatorname{Cov}\left(Z_{B_0,t}, Z_{B_0,k+s}\right) \\
= & \operatorname{Cov}\left(\frac{1}{N}\sum_{i=1}^{N}\mathrm{MMD}^2(X_i^{(B_0,t)}, Y^{(B_0,t)}), \frac{1}{N}\sum_{j=1}^{N}\mathrm{MMD}^2(X_j^{(B_0,t+s)}, Y^{(B_0,t+s)})\right) \\
= & \binom{B_0}{2}^{-2}\binom{(B_0-s)\vee 0}{2}\Big[\frac{1}{N}\mathrm{Var}(h(x,x',y,y')) \\
& + \frac{N-1}{N}\mathrm{Cov}(h(x,x',y,y'), h(x'',x''',y,y'))\Big].
\end{aligned}
$$

Finally, plugging in the expressions for $\mathrm{Var}[Z_{B_0,t}]$ and $\mathrm{Var}[Z_{B_0,t+s}]$, we have (A.26) for the online case. ■

A.3 Proof of Theorem 2

Below, we present the main steps in proving Theorem 2, including (1) exponential tilting; (2) change-of-measure by the likelihood identity; (3) establish properties of the local field and the global term; and (4) perform asymptotic approximation using the localization theorem (Theorem 5.1 in [123] and Sec. 3.4 in [11]) by showing that the "global" log likelihood and the "local process" are asymptotically independent. Finally, we collect terms together to obtain the result.

A.3.1 Step One: Exponential tilting

We first introduce exponential tilting, which creates a family of distributions that is related to the original distribution of Z'_B. Let the log moment generating function of Z'_B be

$$\psi(\theta) = \log \mathbb{E}[e^{\theta Z'_B}]. \tag{A.6}$$

Define a family of new measures

$$d\mathbb{P}_B = \exp\{\theta Z'_B - \psi(\theta)\}\, d\mathbb{P}, \tag{A.7}$$

where $\mathbb{P}$ represents the original probability measure of Z'_B under the null distribution P, $\mathbb{P}_B$ is the new measure after the transformation, and θ parameterizes the family of the new measures. Note that the new measures take the form of exponential family, with θ being the parameter.

Recall that, under the null distribution, Z'_B has zero mean and unit variance. Given the assumption that Z'_B is a standard Gaussian random variable, the corresponding log moment generating function is given by $\psi(\theta) = \theta^2/2$. One has the freedom to select the value of θ to determine the new measure. We will set θ such that the mean under the tilted measure is equal to a given threshold b. This means that the new measure peaks at the threshold b, which enables us to use the local central limit theorem later on. This can be done by choosing θ such that $\dot{\psi}(\theta) = b$, and therefore $\theta = b$. Note that the solution θ does not depend on B. Hence, we can set the mean under the transformed measure to b, by uniformly choosing $\theta = b$ for any B. Given such a choice, the transformed measure is given by $d\mathbb{P}_B = \exp\{bZ'_B(x) - b^2/2\}\, d\mathbb{P}$. We also define, for each B, the log-likelihood ratio $\log(d\mathbb{P}_B/d\mathbb{P})$ of the form

$$\ell_B = bZ'_B - b^2/2. \tag{A.8}$$

This way, we have associated the detection statistic Z'_B with a likelihood ratio, even if Z'_B itself does not come out of a likelihood ratio.

The following lemma shows that Z'_B under the new measure has the same unit variance and its mean has been shifted to b. This key fact will lead to the desired exponential tail.

Lemma A.3.1 (Mean and variance under tilted measure) *Define $\mathbb{E}_B$ and Var$_B$ as the expectation and variance under the transformed measures*

$$\mathbb{E}_B[U] = \mathbb{E}[Ue^{\ell_B}], \tag{A.9}$$

$$\mathrm{Var}_B[U] = \mathbb{E}[U^2 e^{\ell_B}] - \mathbb{E}_B^2[U]. \tag{A.10}$$

We have $\mathbb{E}_B[Z'_B] = b$, and $\mathrm{Var}_B[Z'_B] = 1$.

Proof First, $\mathbb{E}_B[Z'_B] = \dot{\psi}(b) = b$ by construction. To show $\mathrm{Var}_B[Z'_B] = 1$, note that $\log \mathbb{E}[e^{bZ'_B}] = b^2/2$. Taking the derivative of $\psi(\theta)$ with respect to b twice gives $\mathbb{E}[(Z'_B)^2 e^{bZ'_B}] = e^{b^2/2} + b^2 e^{b^2/2}$. Hence, $\mathbb{E}_B[(Z'_B)^2] = \mathbb{E}[(Z'_B)^2 e^{\theta Z'_B - \psi(b)}] = 1 + b^2$, and $\mathrm{Var}_B[Z'_B] = \mathbb{E}_B[(Z'_B)^2] - b^2 = 1$. ■

The following lemma shows that Z'_B under the new measure has the same unit variance with the mean shifted to b. This key fact will lead to the desired exponential tail.

Lemma A.3.2 (Mean and variance under tilted measure) *Define $\mathbb{E}_B$ and Var$_B$ as the expectation and variance under the transformed measures*

$$\mathbb{E}_B[U] = \mathbb{E}[Ue^{\ell_B}], \tag{A.11}$$

$$\mathrm{Var}_B[U] = \mathbb{E}[U^2 e^{\ell_B}] - \mathbb{E}_B^2[U]. \tag{A.12}$$

We have $\mathbb{E}_B[Z'_B] = b$*, and* $\mathrm{Var}_B[Z'_B] = 1$.

Proof First, $\mathbb{E}_B[Z'_B] = \dot{\psi}(b) = b$ by construction. To show $\mathrm{Var}_B[Z'_B] = 1$, note that $\log \mathbb{E}[e^{bZ'_B}] = b^2/2$. Taking the derivative of $\psi(\theta)$ with respect to b twice gives $\mathbb{E}[(Z'_B)^2 e^{bZ'_B}] = e^{b^2/2} + b^2 e^{b^2/2}$. Hence, $\mathbb{E}_B[(Z'_B)^2] = \mathbb{E}[(Z'_B)^2 e^{\theta Z'_B - \psi(b)}] = 1 + b^2$, and $\mathrm{Var}_B[Z'_B] = \mathbb{E}_B[(Z'_B)^2] - b^2 = 1$. ■

A.3.2 Step Two: Change-of-measure

Now we are ready to analyze the tail probability $\mathbb{P}\left\{\max_{2 \le B \le B_{\max}} Z'_B > b\right\}$. The basic idea is to convert the original problem of finding the small probability that the maximum of a random field exceeds a large threshold to another problem: finding an alternative measure under which the event happens with a much higher probability.

Here, the alternative measure will be a mixture of simple exponential tilted measures. Define the maximum and the sum for likelihood ratio differences relative to a particular parameter value B:

$$M_B = \max_{s \in \{2,\ldots,B_{\max}\}} e^{\ell_s - \ell_B}, \qquad S_B = \sum_{s \in \{2,\ldots,B_{\max}\}} e^{\ell_s - \ell_B}. \tag{A.13}$$

Also define a re-centered likelihood ratio, which we call the *global term*

$$\tilde{\ell}_B = b(Z'_B - b).$$

With the definitions above and the log likelihood ratios ℓ_B in (A.8), we have the following

$$\begin{aligned}
&\mathbb{P}\left\{\max_{2\le B\le B_{\max}} Z'_B > b\right\} = \mathbb{E}\left[1; \max_{2\le B\le B_{\max}} Z'_B > b\right] = \mathbb{E}\left[\underbrace{\frac{\sum_{B=2}^{B_{\max}} e^{\ell_B}}{\sum_{s=2}^{B_{\max}} e^{\ell_s}}}_{=1}; \max_{2\le u\le B_{\max}} Z'_u > b\right] \\
&= \sum_{B=2}^{B_{\max}} \mathbb{E}\left[\frac{e^{\ell_B}}{\sum_s e^{\ell_s}}; \max_{2\le u\le B_{\max}} Z'_u > b\right] \overset{(A.11)}{=} \sum_{B=2}^{B_{\max}} \mathbb{E}_B\left[\frac{1}{\sum_s e^{\ell_s}}; \max_{2\le u\le B_{\max}} Z'_u > b\right] \\
&= e^{-b^2/2} \sum_{B=2}^{B_{\max}} \mathbb{E}_B\left[\frac{M_B}{S_B} e^{-(\tilde{\ell}_B + \log M_B)}; \tilde{\ell}_B + \log M_B \ge 0\right]
\end{aligned} \tag{A.14}$$

where an intermediate step is done by changing the measure to $\mathbb{P}_B$, and the last equality can be verified by simple algebra. Recall our notation $\mathbb{E}_B[\mathcal{A}; \mathcal{B}] = \mathbb{E}_B[\mathcal{A}\mathbf{1}\{\mathcal{B}\}]$ for a random quantity $\mathcal{A}$ and event $\mathcal{B}$; $\mathbf{1}$ denotes an indicator function.

In a nutshell, the last equation in (A.14) converts the tail probability to a product of two terms: a deterministic term $e^{-b^2/2}$ associated with the large deviation rate, and a sum of conditional expectations under the transformed measures. A close examination of the conditional expectations of the form $\mathbb{E}_B[\cdots; [\cdots] \ge 0]$ reveals that it involves a product of the ratio M_B/S_B, and an exponential function that depends on $\tilde{\ell}_B$, which plays the role of weight. Under the new measure $\mathbb{P}_B$, $\tilde{\ell}_B$ has zero mean and variance equal to b^2 (shown below in Lemma A.3.3) and it dominates the other term $\log M_B$ and, hence, the probability of exceeding zero will happen with much higher probability. Next, we characterize the limiting ratio and the other factors precisely, by the localization theorem.

A.3.3 Step Three: Establish properties of local and global terms

In (A.14), our target probability has been decomposed into terms that only depend on (i) the *local field* $\{\ell_s - \ell_B\}$, $2 \le s \le B_{\max}$, which are the differences between the log-likelihood ratio with parameter B and with other parameter values s, $2 \le s \le B_{\max}$, and (ii) the *global term* $\tilde{\ell}_B$, which is the centered and scaled likelihood ratio with parameter B. We need to first establish some useful properties of the local field and the global term under the tilted measure. We will eventually show that the local field and the global term are asymptotically independent.

The following property for the global term can be derived from Lemma A.3.2. The result shows that under the tilted measure, the global term $\tilde{\ell}_B$ has zero mean for any B, with variance diverging with b.

Lemma A.3.3 (Global term for offline scan B-statistic) *The mean and variance of the global term $\tilde{\ell}_B = b(Z'_B - b)$, for $2 \leq B \leq B_{\max}$, are given by*

$$\mathbb{E}_B[\tilde{\ell}_B] = 0, \quad \mathrm{Var}_B[\tilde{\ell}_B] = b^2. \tag{A.15}$$

Assuming Z'_B is approximately normal, the local field $\ell_s - \ell_B$ (or equivalently $b(Z'_s - Z'_B)$) and the global term $\tilde{\ell}_B$ (or equivalently $b(Z'_B - b)$) are also approximately normally distributed.

Lemma A.3.4 (Local field for offline scan B-statistic) *The mean and variance of the local field $\{\ell_s - \ell_B\}$, for $|s - B| = 0, 1, 2, \ldots$, are given by*

$$\mathbb{E}_B[\ell_s - \ell_B] = -b^2(1 - r_{s,B}), \quad \mathrm{Var}_B[\ell_s - \ell_B] = 2b^2(1 - r_{s,B}),$$

with $r_{s,B}$ defined in (2.10). For any s_1 and s_2, the covariance between two local field terms is given by

$$\mathrm{Cov}_B\left(\ell_{s_1} - \ell_B, \ell_{s_2} - \ell_B\right) = b^2\left(1 + r_{s_1,s_2} - r_{s_1,B} - r_{s_2,B}\right).$$

Proof Note that $\ell_s - \ell_B = b(Z'_s - Z'_B)$, $\mathbb{E}_B[Z'_B] = b$, $\mathrm{Var}_B[Z'_B] = 1$. Moreover, due to the normal assumption of Z'_B, we have the following decomposition $\mathbb{E}_B[\ell_s - \ell_B] = \mathbb{E}_B[b(Z'_s - Z'_B)] = \mathbb{E}_B[b(r_{s,B}Z'_B + (1 - r^2_{s,B})^{1/2}W - Z'_B)] = -b^2(1 - r_{s,B})$, where W is a zero-mean random variable and independent of Z'_B, representing residual of regression.

The variance and covariance can be found using similar decompositions. ■

Remark A.3.1 (Consequence of Lemma A.3.4) *From the expression of the covariance in (2.10), we have that for $s - B > 0$,*

$$r_{s,B} = [1 + (s - B)/B]^{-1/2} [1 + (s - B)/(B - 1))]^{-1/2},$$

and for $s - B < 0$,

$$r_{s,B} = [1 + (s - B)/B]^{1/2} [1 + (s - B)/(B - 1)]^{1/2} .$$

Consequently,

1. *When $|s-B| \to \infty$, $r_{s,B} \to 0$. Therefore, when $|s-B| \to \infty$, $\mathbb{E}_B[\ell_s - \ell_B]$ converges to $-b^2$ and $\mathrm{Var}_B[\ell_s - \ell_B]$ converges $2b^2$.*

2. *When $|s - B|$ is small, assume $s = B + j$, $j = 0, \pm 1, \pm 2, \ldots$. Perform the Taylor expansion of $r_{B+j,B}$ around 0, we have that*

$$r_{B+j,B} = 1 - \frac{1}{2}\frac{2B-1}{B(B-1)}|j| + o(|j|). \tag{A.16}$$

Define

$$\mu = b\{(2B - 1)/[B(B - 1)]\}^{1/2}. \tag{A.17}$$

Note that μ depends on the threshold as well as B, the block size parameter. Using

(A.16), we have

$$\lim_{|j|\to 0} \mathbb{E}_B[\ell_{B+j} - \ell_B] = -\frac{\mu^2}{2}|j|,$$

$$\lim_{|j|\to 0} \mathrm{Var}_B[\ell_{B+j} - \ell_B] = \mu^2|j|,$$

$$\lim_{|j_1|\to 0, |j_2|\to 0} \mathrm{Cov}_B\left(\ell_{B+j_1} - \ell_B, \ell_{B+j_2} - \ell_B\right) = \mu^2(|j_1| \wedge |j_2|).$$

Therefore, when $|j|$ is small (i.e., in the neighborhood of zero), we can approximate the local field using a two-sided Gaussian random walk with drift $\mu^2/2$ and the variance of the increment being μ^2:

$$\ell_{B+j} - \ell_B \overset{d}{=} \mu \sum_{i=1}^{|j|} \vartheta_i - \mu^2 j/2, \quad j = \pm 1, \pm 2, \ldots \tag{A.18}$$

where ϑ_i are i.i.d. standard normal random variables.

A.3.4 Step Four: Approximation using Localization Theorem

The remaining work is to compute the conditional expectations $\mathbb{E}_B[\cdots; (\cdots) \geq 0]$ for each B in (A.14). In the following, we drop the subscript B in $\mathbb{E}_B$ for simplicity, and the approximation results hold for each B. We assume $b \to \infty$, $B_{\max} \to \infty$, and $b^2/B_{\max}$ is held to a fixed positive constant. Introduce an abstract index κ and let $\kappa = b^2$; this choice is because $\kappa^{1/2}$ is the multiplicative factor that balances the rate of convergence of the global term under the transformed measure. Typically, κ is equal to the variance of the global term $\tilde{\ell}_B = b(Z'_B - b)$, which is b^2 as shown in Lemma A.3.3; κ is also associated with the drift and the variance of the incremental of the local field $\{\ell_s - \ell_B\}$ for $|s - B| = 0, 1, 2, \ldots$, as shown in Lemma A.3.4.

Using a powerful localization theorem (see Theorem 3.1 in [123] or Theorem 5.2 in [11]), we can obtain the limit for each term in the summand of (A.14), rewritten as (by changing the index to κ)

$$\mathbb{E}\left[\frac{M_\kappa}{S_\kappa} e^{-(\tilde{\ell}_\kappa + \log M_\kappa)}; \tilde{\ell}_\kappa + \log M_\kappa \geq 0\right], \tag{A.19}$$

when $\kappa \to \infty$. Basically, the localization theorem states that (A.19) scaled by $\kappa^{\frac{1}{2}}$ converges under mild conditions when $\kappa \to \infty$.

The statement of the theorem involves a local σ-algebra denoted as $\widehat{\mathcal{F}}_\kappa$:

$$\widehat{\mathcal{F}}_\kappa = \sigma\{\ell_s - \ell_B : |s - B| \leq g(\kappa)\}, \tag{A.20}$$

where a function $g(\kappa)$ specifies the size of the local region. The choice of $g(\kappa)$ is critical and it guarantees subsequent convergence. Following the analysis of scan statistics in [11], we choose $g(\kappa) = cb^{-2}$ for some large constant c. This local σ-field is asymptotically independent of $\tilde{\ell}_\kappa$, and it carries all information needed to construct the local field.

Define $\widehat{M}_\kappa$ and $\widehat{S}_\kappa$ as the maximization and summation restricted to a smaller subset of parameter values $\{s : |s - B| \leq g(\kappa)\}$, and they are measurable with respect to $\widehat{\mathcal{F}}_\kappa$. Note that $\widehat{M}_\kappa$ and $\widehat{S}_\kappa$ serve as approximations to M_κ and S_κ. In the limit, the local random field converges to a Gaussian random field, and the ratio $\mathbb{E}[\widehat{M}_\kappa/\widehat{S}_\kappa]$ converges to a limit that can be determined with the parameters of the Gaussian random field.

The localization theorem (Theorem 5.1 in [123] and Sec. 3.4 in [11]) consists of the five conditions as follows.

Theorem A.3.1 (Localization Theorem) *Given $\epsilon > 0$, if for all large κ, all following conditions hold*

I. Both $0 < M_\kappa \leq S_\kappa < \infty$ and $0 < \widehat{M}_\kappa \leq \widehat{S}_\kappa < \infty$ hold in probability one.

II. Denote $A^c = \{|\log M_\kappa - \log \widehat{M}_\kappa| > \epsilon\} \cup \{|\widehat{S}_\kappa/S_\kappa - 1| > \epsilon\}$. For some $0 < \delta$ that *does not depend on ϵ:*

$$\max_{|x|\leq 3g(\kappa)} \mathbb{P}\left[A^c \cap \{\tilde{\ell}_\kappa + \log \widehat{M}_\kappa \in x + (0,\delta]\} \cap \{|\hat{m}| \leq g(\kappa)\}\right] \leq \epsilon \kappa^{-1/2},$$

where $\hat{m}_\kappa = \min\{\log \widehat{M}_\kappa, g(\kappa)\} - \log(1-\epsilon)$.

III. $\mathbb{E}[\widehat{M}_\kappa / \widehat{S}_\kappa]$ *converges to a finite and positive limit denoted by* $\mathbb{E}[M/S]$.

IV. *There exist* $\mu_\kappa \in \mathbb{R}$ *and* $\sigma_\kappa \in \mathbb{R}^+$ *such that for every* $0 < \epsilon'$, δ, *for any event* $E \in \widehat{\mathcal{F}}_\kappa$ *and for all large enough* κ

$$\sup_{|x|\leq \epsilon\kappa^{1/2}} \left| \kappa^{1/2}\mathbb{P}(\tilde{\ell}_\kappa \in x + (0,\delta], E) - \frac{\delta}{\sigma}\phi\left(\frac{\mu}{\sigma}\right)\mathbb{P}(E)\right| \leq \epsilon'.$$

V. $\mathbb{P}(|\log M_\kappa| > \epsilon\kappa^{1/2})$, $\mathbb{P}(|\log \widehat{M}_\kappa| > \epsilon\kappa^{1/2})$ *and* $\mathbb{P}(\log M_\kappa - \log \widehat{M}_\kappa < -\epsilon)$ *are all* $o(\kappa^{-1/2})$.

Then

$$\lim_{\kappa\to\infty} \kappa^{1/2}\mathbb{E}\left[\frac{M_\kappa}{S_\kappa} e^{-\left[\tilde{\ell}_\kappa + \log M_\kappa\right]}; \tilde{\ell}_\kappa + \log M_\kappa \geq 0\right] = \sigma^{-1}\phi\left(\frac{\mu}{\sigma}\right)\mathbb{E}[M/S], \tag{A.21}$$

where $\phi(\cdot)$ *is the density of the standard normal distribution.*

Intuitively, the localization theorem says the following. To find the desired limit of (A.19) as $\kappa \to \infty$, one first approximates M_κ and S_κ by their localized versions, which are obtained by restricting the maximization and summation in a neighborhood of parameter values. Then one can show that the localized ratio M_κ/S_κ is asymptotically independent of the global term $\tilde{\ell}_\kappa$ as $\kappa \to \infty$. The asymptotic analysis is then performed on the local field and the global term separately. The expected value of the localized ratio $\mathbb{E}[M_\kappa/S_\kappa]$ converges to a constant independent of κ, and the limiting conditional distribution of $\tilde{\ell}_\kappa$ can be found using the local central limit theorem. Thus, one can calculate the remaining conditional expectation involving $\tilde{\ell}_\kappa$.

Checking conditions. Let us now verify the validity of the conditions in our setting. First, *Condition I* is met since for Gaussian random variables, $M_\kappa > 0$, $S_\kappa > 0$ with probability 1, and the maximization of a collection of non-negative numbers is smaller or equal to the summation. Similar arguments hold for their counterparts $\widehat{M}_\kappa > 0$ and $\widehat{S}_\kappa > 0$ when the maximization and summation are over a smaller set.

Condition II describes that the localized versions $\widehat{M}_\kappa$ and $\widehat{S}_\kappa$ are good approximations of M_κ and S_κ when κ is sufficiently large, for properly defined $\widehat{\mathcal{F}}_\kappa$. In Section 3.4.4 of [11], the corresponding Condition II has been rigorously checked, assuming a local region defined in the same form of our local region and assuming Gaussian random field. Thus, checking Condition II for our case will follow the same steps, using the properties established in Section A.3.3. We omit the details here.

Condition III is checked by applying the distributional approximations to the localized version of M_κ/S_κ. We can show that the expectation of the ratio $\mathbb{E}[\widehat{M}_\kappa/\widehat{S}_\kappa]$ converges to a finite and positive limit denoted by $\mathbb{E}[M/S]$, which does not depend on κ. Since the increment $\ell_{B+j} - \ell_B$ has negative mean as shown in Lemma A.3.4, the values of M_κ and S_κ will be determined by values j close to 0, so is the ratio M_κ/S_κ. This implies, a relatively small local region centered on B is sufficient.

From Remark A.3.1, the local field when the index is close to the shifted measure parameter B can be approximated as a two-sided Gaussian random walk with drift $-\mu^2/2$ and variance μ^2 (with μ defined in (A.17)), which is denoted as $W(\mu^2 j)$ below. Therefore, we have that with high probability,

$$\mathbb{E}[\widehat{M}_\kappa/\widehat{S}_\kappa] = \mathbb{E}\left[\frac{\max_{|j|\le cb^{-2}} e^{W(\mu^2 j)}}{\sum_{|j|\le cb^{-2}} e^{W(\mu^2 j)}}\right].$$

When $c \to \infty$, it approaches to a limit known as the *Mill's ratio*

$$\mathbb{E}[M/S] = \mathbb{E}\left[\frac{\max_{|j|} e^{W(\mu^2 j)}}{\sum_{|j|} e^{W(\mu^2 j)}}\right],$$

with maximization and summation extending to the entire collection of negative and positive integers. The Mill's ratio is related to the Laplace transform of the overshoot of the maxima of Gaussian random field over a threshold b, and an expression has been obtained based on nonlinear renewal theory (see, [12] and Chapter 2.2 of the book [11]): $\mathbb{E}[M/S] = \exp(-2\sum_{j=1}^{\infty}\Phi(-j^{1/2}\mu/2))$. An easier numerical evaluation is given by $\mathbb{E}[M/S] \approx (\mu^2/2)\nu(\mu)$ for a special function $\nu(\mu)$ defined in (2.9).

Condition IV can be checked via a local multivariate central limit theorem that is local in one component and non-local in others (Theorem 5.3 in [11]). The theorem says the following: assuming ξ_i are independent, identically distributed random vector of dimension $d+1$. Assume the mean of each vector is zero, and variance of the first component converges to a finite σ, the covariance matrix of the last d components converges a finite matrix Σ, and the correlation between these components and the first one converges to zero (hence, the overall covariance matrix is block-diagonal). Define $S_\gamma = \sum_{i=1}^{\gamma}\xi_{i,1}$ and a d dimensional vector with element $h_{\gamma,j} = \gamma^{-1/2}\xi_{i,j}$, for $1 \leq j \leq d$. Then under mild conditions,

$$\lim_{\gamma\to\infty}\gamma^{1/2}\mathbb{P}(S_\gamma \in [l,u], h_\gamma \in \mathcal{A}) = \frac{l-u}{(2\pi)^{1/2}\sigma}\mathbb{P}(h \in \mathcal{A}) \tag{A.22}$$

for any interval $[l,u]$ and an arbitrary set $\mathcal{A}$.

Our setting matches exactly to the above distribution when we set the global term as the first component and the local field as the remaining components. Using the properties in Section A.3.3, we have shown the finite mean and variance (covariance) of the global and local field terms. We only need to show the global term, and the local fields are independent of each other asymptotically. It suffices to prove that the conditional covariance of $\{\ell_{B+j} - \ell_B\}$ given $\tilde{\ell}_B$ converges to the unconditional covariance, and the conditional means converges to the unconditional one. With a slight abuse of notation, $r_1 = r_{B+j_1,B}$ and $r_2 = r_{B+j_2,B}$ and using the linear regression decomposition, when conditioning on Z'_B

(which is proportional to $\tilde{\ell}_B$), the two local field terms are independent of each other:

$$\begin{aligned}&\mathrm{Cov}(b(Z'_{B+j_1} - Z'_B), b(Z'_{B+j_2} - Z'_B)|Z'_B)\\&= \mathrm{Cov}(b(r_1 Z'_B + (1-r_1^2)^{1/2}W_1 - Z'_B), b(r_2 Z'_B + (1-r_2^2)^{1/2}W_2 - Z'_B)|Z'_B) = 0.\end{aligned}$$

where W_1 and W_2 are two mutually independent zero-mean random variables that represent the regression residuals (they are also independent of Z'_B).

On the other hand, using the same decomposition, we can show that without conditioning, the covariance is given by

$$\mathrm{Cov}(b(Z'_{B+j_1} - Z'_B), b(Z'_{B+j_2} - Z'_B)) = b^2(1-r_1)(1-r_2).$$

Hence, when $b \to \infty$, due to the property of local field in equation (A.16), for $|j_1| \leq cb^{-2}$, $|j_2| \leq cb^{-2}$, the unconditioned covariance converges to zero given (A.16), which is equal to the conditioned covariance. Similarly, we can show that the conditional means of $\{Z'_{B+j} - Z'_B\}$ conditioning on Z'_B converges to the unconditional ones.

Now we invoke the local central limit theorem. Since the density of the global term $\tilde{\ell}_B$ is approximately normal, we can calculate a desired form of the probability. From (A.15), the variance of the global term increases with b. The density of $\tilde{\ell}_B$ can be uniformly approximated by $1/(2\pi b^2)^{1/2}$ within a small region around the origin $|x| \leq 3(4/+1+\epsilon)\log b$ [11]. Such an approximation also holds for $\tilde{\ell}_B - x$ given any value x that is not too large. Furthermore, notice that $\log \hat{M}_\kappa$ is very close to 0 and therefore is negligible; this is because $e^{\ell_s - \ell_B}$ should attain its maximal value when $|s-B|$ close to 0 as analyzed before. Let $\mu_\kappa = \mathbb{E}_B[\tilde{\ell}_\kappa/b] = 0$ and $\sigma_\kappa^2 = \mathrm{Var}_B[\tilde{\ell}_\kappa/b] = 1$. When $\kappa = b^2 \to \infty$, using local central limit theorem (A.22), we have that

$$\kappa^{1/2}\mathbb{P}\left(\tilde{\ell}_\kappa \in x - \log \hat{M}_\kappa + (0,\delta]\right) \to \frac{\delta}{\sigma_\kappa}\phi\left(\frac{\mu_\kappa}{\sigma_\kappa}\right). \tag{A.23}$$

Condition V is checked as follows. Note that the terms inside the M_κ are likelihood ratios with unit expectation since $\mathbb{E}_B[\exp(\ell_B)] = 1$. Thus, $\exp(\ell_s - \ell_B)$ is a martingale and by a standard martingale inequality, $\mathbb{P}(\log M_\kappa > \epsilon\kappa^{1/2}) \leq \exp(-\epsilon\kappa^{1/2})$. Then using a similar argument as in [123], one can show the other two inequalities, since $\widehat{M_\kappa}$ is an approximation to M_κ.

Finally, since all conditions are met, we can now apply the localization theorem for $b \to \infty$ and put things together to obtain

$$\mathbb{E}_B\left[\frac{M_B}{S_B}e^{-[\tilde{\ell}_B+\log M_B]}; \tilde{\ell}_B + \log M_B \geq 0\right] = \frac{\mu^2}{2}\nu(\mu)\frac{1}{\sqrt{2\pi b^2}}(1+o(1)). \tag{A.24}$$

Substitute (A.24) back to the likelihood ratio identity (A.14), and we arrive at the approximation in Theorem 2.

A.4 Proof of Theorem 11

The method for approximating the ARL is related to that used to analyze the offline scan B-statistic. In addition, we need the following lemma.

Lemma A.4.1 (Asymptotic null distribution of T) *Under the null, when $b \to \infty$, the stopping time T defined in (2.6) is uniformly integrable and asymptotically exponentially distributed, i.e.,*

$$|\mathbb{P}\{T \geq m\} - \exp(-\lambda_0 m)| \to 0,$$

in the range where $m\lambda_0$ is bounded away from 0.

Proof The proof is based on adapting arguments in [124, 125, 126]. The main idea is to show that the number of boundary cross events for detection statistic over disjoint intervals converges to Poisson random variable in the total variation norm, resulted from the Poisson limit theorem (Theorem 1 in [127]) for dependent samples. First, we show that the stopping time T is asymptotically exponentially distributed. The analysis of the distribution of the stopping time is based on Poisson approximation. Define an indicator of the event $\mathbf{1}_j$ such that the event $\mathbf{1}\{\max_{(j-1)m\leq t\leq jm} Z'_{B_0,t} > b\}$. Consider the time interval $[0,x]$. Note that the stopping time is not activated in the interval $[0,x]$, if and only if, all the relevant indicators are zero. For simplicity, we assume x is divisible by m. Define the random variable $\widehat{W} = \sum_{j=1}^{x/m} \mathbf{1}_j$. Hence, $\{\widehat{W} = 0\} = \{T_b > x\}$. Thus, to characterize the tail probability of the stopping time $\mathbb{P}\{T_b > x\}$, we show that the sum of the indicator functions converge to a Poisson distribution. ■

Using Lemma A.4.1, we know for large m, $\mathbb{P}\{T \leq m\}$ is approximately $1-\exp(-\lambda_0 m) \approx \lambda_0 m$, and $\mathbb{E}\{T\}$ is equal to λ_0^{-1} asymptotically when $b \to \infty$. So the remaining question is to find the probability and the corresponding λ_0. Consider $\mathbb{P}\{T \leq m\} = \mathbb{P}\{\max_{2\leq t\leq m} Z'_{B_0,t} > b\}$. Suppose $m > B_0$ and $\log b \ll m \ll b^{-1}e^{\frac{1}{2}b^2}$. We will adopt a similar strategy to approximate this probability using the change-of-measure technique.

Note that the covariance structures for online and offline scan B-statistics are different, so there will be different drift parameters when we invoke the localization theorem. Using exponential tilting, we introduce a likelihood ratio

$$\zeta_t = bZ'_{B_0,t} - b^2/2.$$

Again using the change-of-measure by likelihood ratio identity, we obtain

$$\mathbb{P}\left\{\max_{2\leq t\leq m} Z'_{B_0,t} > b\right\} = e^{-b^2/2}\sum_{t=2}^{m} \mathbb{E}_t\left[\frac{M'_t}{S'_t} e^{-[\tilde{\zeta}_t + \log M_t]}; \tilde{\zeta}_t + \log M'_t \geq 0\right], \qquad \text{(A.25)}$$

where

$$M'_t = \max_{2\leq s\leq m} e^{\zeta_s - \zeta_t}, \quad S'_t = \sum_{2\leq s\leq m} e^{\zeta_s - \zeta_t}, \quad \text{and} \quad \tilde{\zeta}_t = b(Z'_{B_0,t} - b).$$

Hence, one can again apply the localization theorem to find the approximation when $b \to \infty$, and the only differences are in the definition and characterization of global and local field terms.

Lemma A.4.2 (Local field of online scan B-statistic) *The mean, variance, and covariance of the local field $\{\zeta_s - \zeta_t\}$ are given by*

$$\mathbb{E}_t[\zeta_s - \zeta_t] = -b^2(1-\rho_{s,t}), \quad \mathrm{Var}_t[\zeta_s - \zeta_t] = 2b^2(1-\rho_{s,t}),$$
$$\mathrm{Cov}_t\left(\zeta_{s_1} - \zeta_t, \zeta_{s_2} - \zeta_t\right) = b^2\left(1 + \rho_{s_1,s_2} - \rho_{s_1,t} - \rho_{s_2,t}\right),$$

where

$$\rho_{s,t} = \mathrm{Cov}(Z'_{B_0,s}, Z'_{B_0,t}) = \frac{\binom{(B_0-|t-s|)\vee 0}{2}}{\binom{B_0}{2}}. \tag{A.26}$$

The proof can be found in Appendix A.2.3. Note that when $|t-s|$ is close to 0, $\mathbb{E}_t[\zeta_s - \zeta_t]$ is close to 0. With an increasing $|t-s|$, $\mathbb{E}_t[\zeta_s - \zeta_t]$ decreases until $|t-s| > B_0$ (when there are no overlapping test data in the sliding block), then $\mathbb{E}_t[\zeta_s - \zeta_t]$ becomes $-b^2$. The values of M_κ and S_κ as in localization theorem will be determined by the values of $|j|$ close to 0.

Now, again, we will use an argument based on Taylor expansion to find the drift term of the local field. When $|s-t|$ is close to 0, we can approximate $\{\zeta_s - \zeta_t\}$ as a two-sided random walk. Using Taylor expansion, we have

$$\rho_{t+j,t} = 1 - \frac{2B_0 - 1}{B_0(B_0-1)}|j| + o(|j|). \tag{A.27}$$

Let $\lambda = b[2(2B_0-1)]/[B_0(B_0-1)]^{1/2}$. Hence, we can show that the mean, variance, and covariance of the local field are approximately

$$\lim_{|j|\to 0} \mathbb{E}_t[\zeta_{t+j} - \zeta_t] = -\frac{\lambda^2}{2}|j|,$$
$$\lim_{|j|\to 0} \mathrm{Var}_t[\zeta_{t+j} - \zeta_t] = \lambda^2|j|,$$
$$\lim_{|j_1|\to 0, |j_2|\to 0} \mathrm{Cov}_t\left(\zeta_{t+j_1} - \zeta_t, \zeta_{t+j_2} - \zeta_t\right) = \lambda^2(|j_1| \wedge |j_2|).$$

As a result, by invoking the localization theorem through a similar set of steps, we obtain

$$\mathbb{P}\{T \le m\} = m \cdot \frac{be^{-\frac{1}{2}b^2}}{\sqrt{2\pi}} \frac{(2B_0-1)}{B_0(B_0-1)} \cdot \nu\left(b\sqrt{\frac{2(2B_0-1)}{B_0(B_0-1)}}\right)(1+o(1)), \tag{A.28}$$

Matching this to above, we know λ_0 is the factor that multiplies m and this leads to the desired result.

For online scan B-statistics, the standard Poisson limit cannot be directly applied, since the events $\{\mathbf{1}_j\}$, $j = 1, \ldots, x/m$, are not independent, and we need the generalized Poisson limit theorem [127], which allows for dependence between the variables. The setup for the theorem is as follows. Let I be an arbitrary index set, and for $\alpha \in I$, let X_α be a Bernoulli random variable with $p_\alpha = \mathbb{P}(X_\alpha = 1) > 0$. Let $W = \sum_{\alpha \in I} X_\alpha$. For each $\alpha \in I$, suppose we choose $B_\alpha \subset I$ with $\alpha \in B_\alpha$. Think of B_α as a "neighborhood of dependence" for each α, such that X_α is independent or nearly independent of all of the X_β for $\beta \notin B_\alpha$. Define $p_1 = \sum_{\alpha\in I}\sum_{\beta\in B_\alpha} p_\alpha p_\beta$, $p_2 = \sum_{\alpha_I}\sum_{\alpha\neq\beta\in B_\alpha} \mathbb{E}(X_\alpha X_\beta)$, $p_3 = \sum_{\alpha\in I} \mathbb{E}|\mathbb{E}(X_\alpha - p_\alpha|\sigma(X_\beta : \beta \in I - B_\alpha))|$, where $\sigma(\cdot)$ represents the σ-field generated by the corresponding random field. Loosely speaking, p_1 measures the neighborhood size, p_2 measures the expected number of neighbors of a given occurrence and p_3 measures the dependence between an event and the number of occurrences outside its neighborhood. Then, we have the following theorem.

Theorem A.4.1 (Poisson approximation, Theorem 1 in [127]) *Let W be the number of occurrences of dependent events, and let Z be a Poisson random variable with $\mathbb{E}Z = \mathbb{E}W = \lambda > 0$. Then the total variation distance between the distributions of W and Z is bounded by*

$$\sup_{\|h\|=1} |\mathbb{E}h(W) - \mathbb{E}h(Z)| \le p_1 + p_2 + p_3.$$

where $h : \mathbb{Z}^+ \to \mathbb{R}$, $\|h\| = \sup_{k\ge 0}|h(k)|$.

The theorem is a consequence of the powerful Chen-Stein method.

Invoking the above theorem in our online scan B-statistics setting, we can bound the total variation distance between the random variable, defined as the number of boundary cross events for the statistic over disjoint intervals, and a Poisson random variable with the same rate. In our setting, let $I = \{1, 2, \ldots, x/m\}$ and $\mathcal{N}(j) = \{j-1, j, j+1\}$ where $j = 2, \ldots (x/m - 1)$ (with obvious modifications for $j = 1$ and $j = x/m$). Then we can specify:

$$p_1 = \sum_{j \in I} \sum_{i \in \mathcal{N}(j) \backslash \{j\}} \mathbb{P}\{\mathbf{1}_j = 1\} \mathbb{P}\{\mathbf{1}_i = 1\} = 2(x/m - 2)\mathbb{P}\{\mathbf{1}_1 = 1\}^2 + 2\mathbb{P}\{\mathbf{1}_1 = 1\}, \tag{A.29}$$

$$p_2 = \sum_{j \in I} \sum_{i \in \mathcal{N}(j) \backslash \{j\}} \mathbb{P}\{\mathbf{1}_j = 1, \mathbf{1}_i = 1\} = 2(x/m - 1)\mathbb{P}\{\mathbf{1}_1 = 1, \mathbf{1}_2 = 1\}, \tag{A.30}$$

$$p_3 = \sum_{j \in I} \mathbb{E}\left\{|\mathbb{E}\{\mathbf{1}_j | \sigma\{\mathbf{1}_i : i \notin \mathcal{N}(j)\}\} - \mathbb{E}\{\mathbf{1}_j\}|\right\}. \tag{A.31}$$

We will show that p_1, p_2, and p_3 converge to 0 as $b \to \infty$. For p_1, the last summand in (A.29) is associated with the two edge elements. It follows that p_1 is asymptotically to $(2C + 2)\mathbb{P}\{\mathbf{1}_1 = 1\}$, which will converge to zero as $b \to \infty$ since $\mathbb{P}\{\mathbf{1}_1 = 1\}$ converges to zero when m is sub-exponential, i.e., $\log b \ll m \ll b^{-1} e^{\frac{1}{2}b^2}$. Next, let us examine p_2 in (A.30). Redefine parameter sub-region

$$S_1 = [0, m - B_0/2], \quad S_2 = [m - B_0/2, m + B_0/2], \quad S_3 = [m + B_0/2, 2m],$$

and denote Y_i, $i = 1, 2, 3$ as $\{Y_i = 1\} = \{\max_{t \in S_i} Z'_{B_0,t} > b\}$, which are the indicator functions of crossings of the threshold in the approximate sub-regions. Notice that the indicator functions Y_1 and Y_3 are independent of each other and they share the same distribution. We use the fact that unless the crossing occurs in a shared sub-region, it must simultaneously occur in two disjoint sub-regions in order to have double crossing. As a consequence, we obtain the upper bound $\mathbf{1}_1 \cdot \mathbf{1}_2 \leq Y_2 + Y_1 \cdot Y_3$, and

$$\mathbb{P}\{\mathbf{1}_1 = 1, \mathbf{1}_2 = 1\} \leq \mathbb{P}\{Y_2 = 1\} + \mathbb{P}\{Y_1 = 1\}^2 \leq \mathbb{P}\{Y_2 = 1\} + \mathbb{P}\{\mathbf{1}_1 = 1\}^2.$$

The probability $\mathbb{P}\{Y_2 = 1\}$ is proportional to $B_0 \cdot be^{-\frac{1}{2}b^2}$. Consequently, p_2 is asymptotically bounded by $2C(B_0/m + \mathbb{P}\{\mathbf{1}_1 = 1\})$. Hence, p_2 converges to zero if $\log b \ll m \ll b^{-1}e^{\frac{1}{2}b^2}$ whenever $b \to \infty$. For p_3 in (A.31), $\mathbf{1}_j$ and $\mathbf{1}_i$ are computed over non-overlapping observations and are therefore independent. Thus, the term p_3 vanishes.

Next prove that the collection of stopping times $\{T_b\}$ indexed by b is *uniformly integrable*. Again consider the sequence of indicators $\{\mathbf{1}_j\}$, $j = 2k$ and $k = 1, 2, \ldots$. Define the random variable τ that identifies the index of the first indicator in the sequence that obtains the value one: $\tau = \inf\{k : \mathbf{1}_{2k} = 1\}$. Note that τ has a geometric distribution. Moreover, since $T_b \leq 2m\tau$ we obtain that

$$\mathbb{P}\{T_b > x\} \leq \mathbb{P}\{\tau > x/(2m)\} = (1 - \mathbb{P}(\mathbf{1}_2 = 1))^{\lfloor x/(2m) \rfloor}.$$

The conclusion then follows from that $1/m \cdot \mathbb{P}(\mathbf{1}_2 = 1)$ converges to 0.

A.5 Skewness correction

In the following, Lemma A.5.1, Lemma A.5.2, and Lemma A.5.3 are used to derive the final expression for the skewness of the scan B-statistic:

Lemma A.5.1 *Under null hypothesis,*

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathit{MMD}^2(X_i, Y)\right)^3\right] \\
=&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x',x'',y',y'')h(x'',x,y'',y)\right] + \frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')^3\right].
\end{aligned}$$

Proof Note that

$$\begin{aligned}\mathbb{E}\left[\left(\text{MMD}^2(X_i,Y)\right)^3\right] =& \binom{B}{2}^{-3}\mathbb{E}\left[\left(\sum_{a<b} h(X_{i,a},X_{i,b},Y_a,Y_b)\right)^3\right]\\ =& \binom{B}{2}^{-3}\sum_k C_k\mathbb{E}\left[h_{ab}h_{cd}h_{ef}\right],\end{aligned}$$

where for simplicity we write $h_{ab} = h(X_{i,a},X_{i,b},Y_a,Y_b)$ and define C_k the corresponding number of combination under specific structure. Most of the terms in $\mathbb{E}\left[h_{ab}h_{cd}h_{ef}\right]$ vanish under the null. By enumerating all the combinations, only two terms are nonzero: $\mathbb{E}\left[h_{ab}h_{bc}h_{ca}\right]$ and $\mathbb{E}\left[h_{ab}h_{ab}h_{ab}\right]$. Then,

$$\begin{aligned}&\mathbb{E}\left[\left(\text{MMD}^2(X_i,Y)\right)^3\right] = \binom{B}{2}^{-3}\binom{B}{2}2(B-2)\mathbb{E}\left[h_{ab}h_{bc}h_{ca}\right] + \binom{B}{2}^{-3}\binom{B}{2}\mathbb{E}\left[h_{ab}h_{ab}h_{ab}\right]\\ =&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(X_{i,a},X_{i,b},Y_a,Y_b)h(X_{i,b},X_{i,c},Y_b,Y_c)h(X_{i,c},X_{i,a},Y_c,Y_a)\right]\\ &+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(X_{i,a},X_{i,b},Y_a,Y_b)^3\right]\\ =&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x',x'',y',y'')h(x'',x,y'',y)\right] + \frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')^3\right].\end{aligned}$$

■

Lemma A.5.2 *Under null hypothesis,*

$$\begin{aligned}&\mathbb{E}\left[\left(\mathit{MMD}^2(X_i,Y)\right)^2\mathit{MMD}^2(X_j,Y)\right]_{i\neq j}\\ =&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x',x'',y',y'')h(x''',x'''',y'',y)\right]\\ &\quad+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')^2h(x'',x''',y,y')\right].\end{aligned}$$

Proof Note that

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathrm{MMD}^2(X_i, Y)\right)^2 \mathrm{MMD}^2(X_j, Y)\right]_{i\neq j}\\
=&\binom{B}{2}^{-3}\mathbb{E}\left[\left(\sum_{a<b} h(X_{i,a}, X_{i,b}, Y_a, Y_b)\right)^2 \left(\sum_{a<b} h(X_{j,a}, X_{j,b}, Y_a, Y_b)\right)\right]\\
=&\binom{B}{2}^{-3}\sum_k C_k \mathbb{E}\left[h_{i,ab}h_{i,cd}h_{j,ef}\right],
\end{aligned}$$

where for simplicity we write $h_{i,ab} = h(X_{i,a}, X_{i,b}, Y_a, Y_b)$ and define C_k the corresponding number of combination under specific structure. Similarly, most of the terms in $\mathbb{E}\left[h_{i,ab}h_{i,cd}h_{j,ef}\right]$ vanish under the null. By enumerating all the combinations, only two terms are nonzero: $\mathbb{E}\left[h_{i,ab}h_{i,bc}h_{j,ca}\right]$ and $\mathbb{E}\left[h_{i,ab}h_{i,ab}h_{j,ab}\right]$. Then,

$$\begin{aligned}
&\mathbb{E}\left[\left(\mathrm{MMD}^2(X_i, Y)\right)^2 \mathrm{MMD}^2(X_j, Y)\right]_{i\neq j}\\
=&\binom{B}{2}^{-3}\binom{B}{2}2(B-2)\mathbb{E}\left[h_{i,ab}h_{i,bc}h_{j,ca}\right] + \binom{B}{2}^{-3}\binom{B}{2}\mathbb{E}\left[h_{i,ab}h_{i,ab}h_{j,ab}\right]\\
=&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(X_{i,a}, X_{i,b}, Y_a, Y_b)h(X_{i,b}, X_{i,c}, Y_b, Y_c)h(X_{j,c}, X_{j,a}, Y_c, Y_a)\right]\\
&+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(X_{i,a}, X_{i,b}, Y_a, Y_b)^2 h(X_{j,a}, X_{j,b}, Y_a, Y_b)\right]\\
=&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x, x', y, y')h(x', x'', y', y'')h(x''', x'''', y'', y)\right]\\
&+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x, x', y, y')^2 h(x'', x''', y, y')\right].
\end{aligned}$$

■

Lemma A.5.3 *Under null hypothesis,*

$$\begin{aligned}
&\mathbb{E}\left[\mathit{MMD}^2(X_i,Y)\mathit{MMD}^2(X_j,Y)\mathit{MMD}^2(X_r,Y)\right]_{i\neq j\neq r}\\
=&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x'',x''',y',y'')h(x'''',x''''',y'',y)\right]\\
&+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x'',x''',y,y')h(x'''',x''''',y,y')\right].
\end{aligned}$$

Proof Note that

$$\begin{aligned}
&\mathbb{E}\left[\text{MMD}^2(X_i,Y)\text{MMD}^2(X_j,Y)\text{MMD}^2(X_r,Y)\right]_{i\neq j\neq r}\\
=&\binom{B}{2}^{-3}\mathbb{E}\left[\left(\sum_{a<b}h(X_{i,a},X_{i,b},Y_a,Y_b)\right)\left(\sum_{c<d}h(X_{j,c},X_{j,d},Y_c,Y_d)\right)\left(\sum_{e<f}h(X_{r,e},X_{r,f},Y_e,Y_f)\right)\right]\\
=&\binom{B}{2}^{-3}\sum_k C_k\mathbb{E}\left[h_{i,ab}h_{j,cd}h_{r,ef}\right].
\end{aligned}$$

Similarly, most of the terms in $\mathbb{E}\left[h_{i,ab}h_{j,cd}h_{r,ef}\right]$ vanish under the null. By enumerating all the combinations, only two terms are nonzero: $\mathbb{E}\left[h_{i,ab}h_{j,bc}h_{r,ca}\right]$ and $\mathbb{E}\left[h_{i,ab}h_{j,ab}h_{r,ab}\right]$. Then,

$$\begin{aligned}
&\mathbb{E}\left[\text{MMD}^2(X_i,Y)\text{MMD}^2(X_j,Y)\text{MMD}^2(X_r,Y)\right]_{i\neq j\neq r}\\
=&\binom{B}{2}^{-3}\binom{B}{2}2(B-2)\mathbb{E}\left[h_{i,ab}h_{j,bc}h_{r,ca}\right]+\binom{B}{2}^{-3}\binom{B}{2}\mathbb{E}\left[h_{i,ab}h_{j,ab}h_{r,ab}\right]\\
=&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(X_{i,a},X_{i,b},Y_a,Y_b)h(X_{j,b},X_{j,c},Y_b,Y_c)h(X_{r,c},X_{r,a},Y_c,Y_a)\right]\\
&+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(X_{i,a},X_{i,b},Y_a,Y_b)h(X_{j,a},X_{j,b},Y_a,Y_b)h(X_{r,a},X_{r,b},Y_a,Y_b)\right]\\
=&\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x'',x''',y',y'')h(x'''',x''''',y'',y)\right]\\
&+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x'',x''',y,y')h(x'''',x''''',y,y')\right].
\end{aligned}$$

■

Using results from Lemma A.5.1, Lemma A.5.2, and Lemma A.5.3, and we can derive the final expression for the skewness of the scan B-statistic, as summarized in Lemma 5.

Proof We can write the raw third-order moment as

$$
\begin{aligned}
&\mathbb{E}[Z_B^3] \\
=&\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\text{MMD}^2(X_i,Y)\right)^3\right] \\
=&\frac{1}{N^3}\mathbb{E}\left[\left(\sum_{i=1}^{N}\text{MMD}^2(X_i,Y)\right)\left(\sum_{j=1}^{N}\text{MMD}^2(X_j,Y)\right)\left(\sum_{r=1}^{N}\text{MMD}^2(X_r,Y)\right)\right] \\
=&\frac{1}{N^3}N\mathbb{E}\left[\left(\text{MMD}^2(X_i,Y)\right)^3\right]+\frac{1}{N^3}\binom{3}{2}\binom{N}{1}\binom{N-1}{1}\mathbb{E}\left[\left(\text{MMD}^2(X_i,Y)\right)^2\text{MMD}^2(X_j,Y)\right]_{i\neq j} \\
&+\frac{1}{N^3}\binom{N}{1}\binom{N-1}{1}\binom{N-2}{1}\mathbb{E}\left[\text{MMD}^2(X_i,Y)\text{MMD}^2(X_j,Y)\text{MMD}^2(X_r,Y)\right]_{i\neq j\neq r} \\
=&\frac{1}{N^2}\left\{\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x',x'',y',y'')h(x'',x,y'',y)\right]+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')^3\right]\right\} \\
&+\frac{3(N-1)}{N^2}\left\{\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x',x'',y',y'')h(x''',x'''',y'',y)\right]\right. \\
&\qquad\left.+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')^2h(x'',x''',y,y')\right]\right\} \\
&+\frac{(N-1)(N-2)}{N^2}\left\{\frac{8(B-2)}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x'',x''',y',y'')h(x'''',x''''',y'',y)\right]\right. \\
&\qquad\left.+\frac{4}{B^2(B-1)^2}\mathbb{E}\left[h(x,x',y,y')h(x'',x''',y,y')h(x'''',x''''',y,y')\right]\right\}
\end{aligned}
$$

■

A.6 Z_B does not converge to Gaussian

Note that the third-order moment of Z_B scales as $\mathcal{O}(B^{-3})$ (due to (2.12)), but when dividing by its variance which scales as $\mathcal{O}(B^{-2})$, the skewness becomes a constant with respect to B. Furthermore, examining the Taylor expansion of moment generating function at $\theta=0$, we have

$$
\mathbb{E}[e^{\theta Z'_B}]=1+\underbrace{\mathbb{E}[Z'_B]}_{0}\theta+\frac{\theta^2}{2}\underbrace{\mathbb{E}[(Z'_B)^2]}_{1}+\frac{\theta^3}{6}\mathbb{E}[(Z'_B)^3e^{\theta Z'_B}]+o(\theta^3).
$$

Recall that the moment generating function of a standard normal Z is given by $\mathbb{E}[e^{\theta Z}] = 1 + \theta^2/2 + o(\theta^3)$. The difference between the two moment generating functions is given by

$$\left|\mathbb{E}[e^{\theta Z'_B}] - \mathbb{E}[e^{\theta Z}]\right| = \frac{|\theta|^3}{6}|\mathbb{E}[(Z'_B)^3 e^{\theta' Z'_B}]| + o(\theta^3) > \frac{|\theta|^3}{6}c|\mathbb{E}[(Z'_B)^3]| + o(\theta^3), \quad \text{(A.32)}$$

where the inequality is due to the fact that $e^{\theta' Z'_B} > 0$ and we may assume it is larger than an absolute constant c. Note that the first term on the right hand side of (A.32) is given by $(c\theta^3/6)\mathrm{Var}[\mathrm{Z_B}]^{-3/2}|\mathbb{E}[\mathrm{Z_B}^3]|$, which is clearly bounded away from zero. Hence,

$$\left|\mathbb{E}[e^{\theta Z'_B}] - (1 + \frac{\theta^2}{2})\right| > \frac{|\theta|^3}{6}\gamma + o(\theta^3)$$

for some constant $\gamma > 0$. This shows that the difference between the moment generating functions of Z'_B and a standard normal is always non-zero and, hence, Z'_B does not converge to a standard normal in any sense. This explains why incorporating the skewness of Z_B can improve the accuracy of the approximations for SL in Theorem 2 and for ARL in Theorem 11.

A.7 More details for real data experiments

A.7.1 CENSREC-1-C speech dataset

CENSREC-1-C is a real-world speech dataset in the Speech Resource Consortium (SRC) corpora provided by National Institute of Informatics (NII)[1]. This dataset contains two categories of data: (1) Simulated data. The simulated speech data are constructed by concatenating several utterances spoken by one speaker. Each concatenated sequence is then added with 7 different levels of noise from 8 different environments. So there are totally 56 different types of noise. Each noise setting contains 104 sequences from 52 males and 52

[1] Available from http://research.nii.ac.jp/src/en/CENSREC-1-C.html

females speakers. (2) Recording data. The recording data is from two real-noisy environments (in university restaurant and in the vicinity of highway), and with two Signal Noise Ratio (SNR) settings (lower and higher). Ten subjects were employed for recording, and each one has four speech sequence data.

Experiment Settings. We will compare our algorithm with the baseline algorithm from [6]. [6] only utilized 10 sequences from "STREET_SNR_HIGH" setting in recording data. Here we will use all the settings in recording data, the SNR level 20 dB and clean signals from simulated data. See Figure A.3 for some examples of the testing data, as well as the statistics computed by our algorithm. The red vertical bar shown in the upper part of each figure is the ground truth of change-point; The green vertical bar shown in the lower part is the change-point detected by our algorithm (the point where the statistic exceeds the threshold). We also plot the threshold as a red dashed horizontal line in each figure. Once the statistics touch the threshold, we will stop the detection. For each sequence, we decompose it into several segments. Each segment consists of two types of signals (noise vs speech). Given the reference data from noise, we want to detect the point where the signal changes from noise to speech.

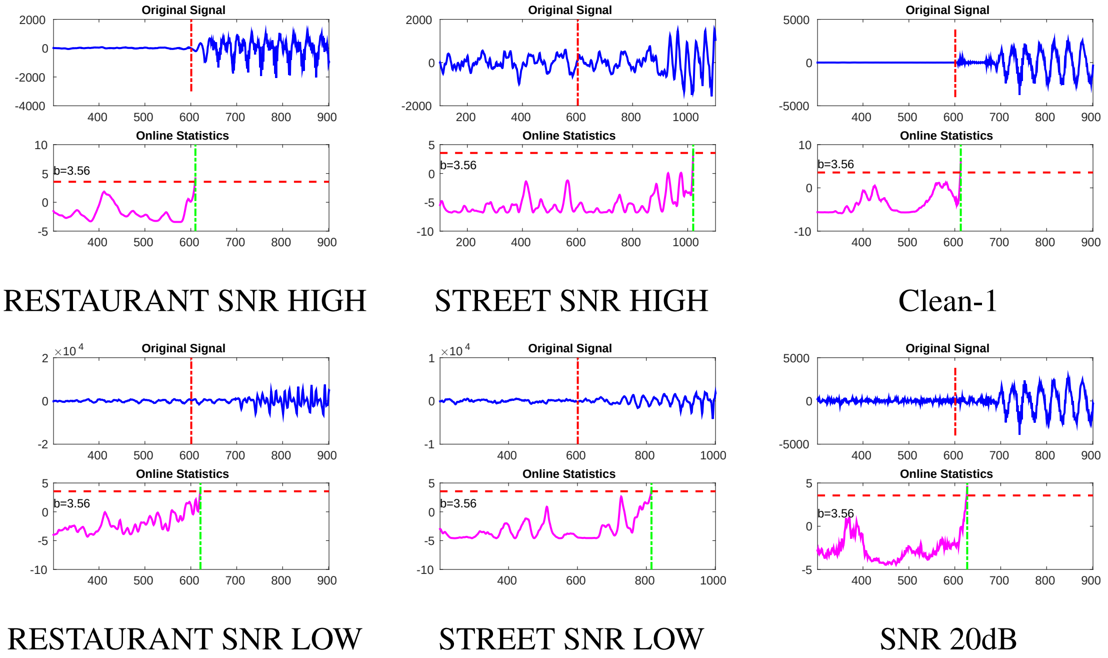
Evaluation Metrics. We use Area Under Curve (AUC) to evaluate the computed statistics, like in [6]. Specifically, for each test sequence that consists of two signal distributions, we will mark the points as change-points whose statistics exceed the given threshold. If the distance between the detected point and true change-point is within the size of detection window, then we consider it as True Alarm (True Positive). Otherwise it is a False Alarm (False Positive).

We use 10% of the sequences to tune the parameters of both algorithms, and use the rest 90% for reporting AUC. The kernel bandwidth is tuned in

$$\{0.1d_{\text{med}}, 0.5d_{\text{med}}, d_{\text{med}}, 2d_{\text{med}}, 5d_{\text{med}}\},$$

where d_{med} is the median of pairwise distances of reference data. Block size is fixed to be 50, and the number of blocks is simply tuned in $\{10, 20, 30\}$.

Results. Table A.7.1 shows the AUC of two algorithms on different background settings. Our algorithm outperforms the baseline on most cases. Both algorithms are performing quite well on the simulated clean data, since the difference between speech signals and background is more significant than the noisy ones. The averaged AUC of our algorithm on all these settings is **.8014**, compared to **.7578** achieved by the baseline algorithm. See the ROC curves in Figure A.4 for a complete comparison.



RESTAURANT SNR HIGH

STREET SNR HIGH

Clean-1

RESTAURANT SNR LOW

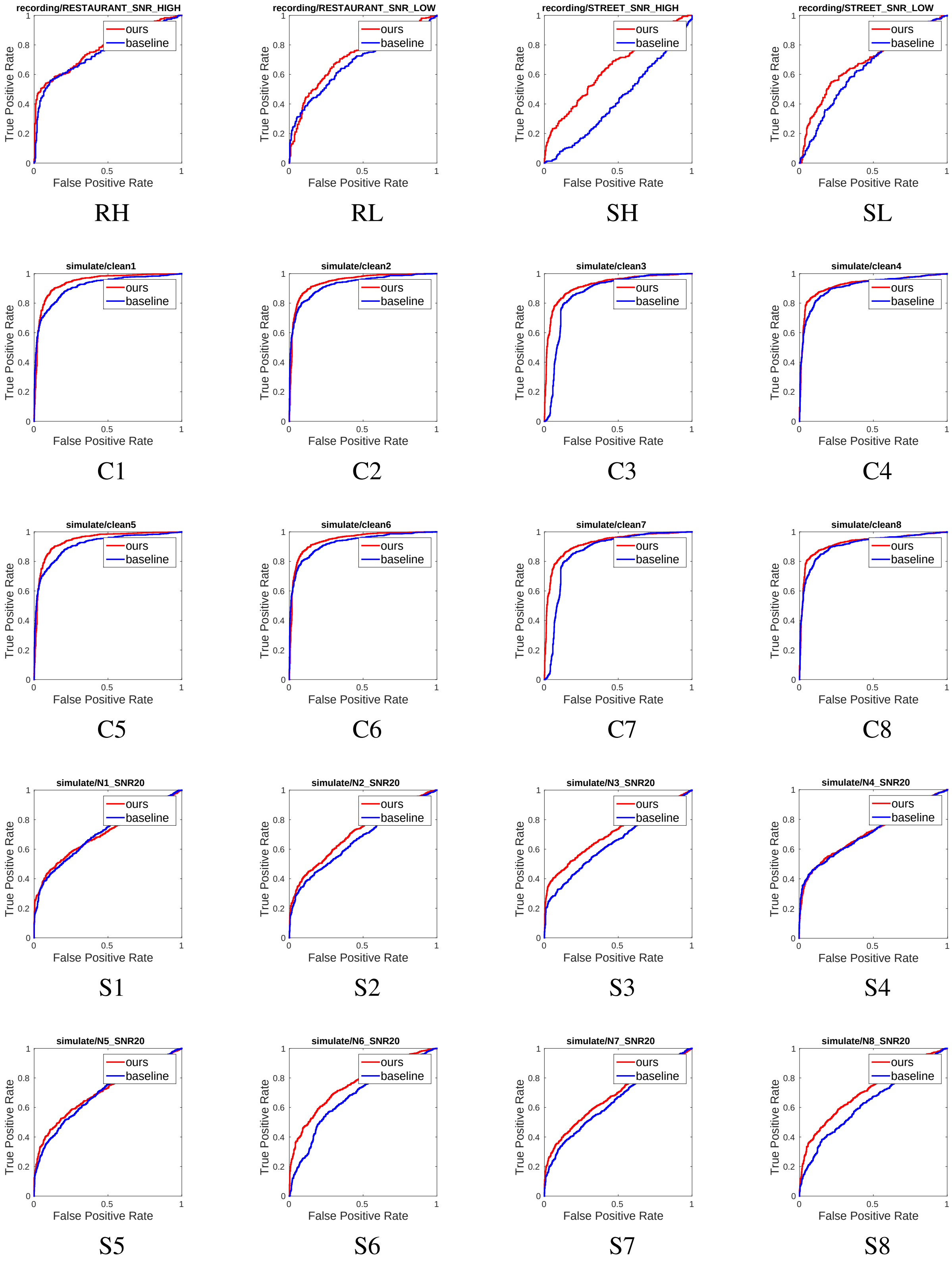
STREET SNR LOW

SNR 20dB

Figure A.3: Examples of speech dataset.

Table A.1: AUC results in CENSREC-1-C speech dataset.

	RH	RL	SH	SL
Ours	**0.7800**	**0.7282**	**0.6507**	**0.6865**
Baseline	0.7503	0.6835	0.4329	0.6432



RH RL SH SL

C1 C2 C3 C4

C5 C6 C7 C8

S1 S2 S3 S4

S5 S6 S7 S8

Figure A.4: ROC curves comparison for speech dataset.

Table A.2: Simulate data with low SNR, with noise from different environment.

	C1	C2	C3	C4	C5	C6	C7	C8
Ours	**0.9413**	**0.9446**	**0.9236**	**0.9251**	**0.9413**	**0.9446**	**0.9236**	**0.9251**
Baseline	0.9138	0.9262	0.8691	0.9128	0.9138	0.9216	0.8691	0.9128

Table A.3: Simulated data with SNR = 20 dB, with noise from different environment.

	S1	S2	S3	S4	S5	S6	S7	S8
Ours	0.7048	**0.7160**	**0.7126**	**0.7129**	**0.7094**	**0.7633**	**0.6796**	**0.7145**
Baseline	**0.7083**	0.6681	0.6490	0.7119	0.6994	0.6815	0.6487	0.6541

A.7.2 HASC human activity dataset

This data set is from *Human Activity Sensing Consortium (HASC) challenge 2011*[2]. Each data consists of human activity information collected by portable three-axis accelerometers. Following the setting in [6], we use the ℓ_2-norm of 3-dimensional data (i.e., the magnitude of acceleration) as the signals.

We use the 'RealWorldData' from HASC Challenge 2011, which consists of 6 kinds of human activities:

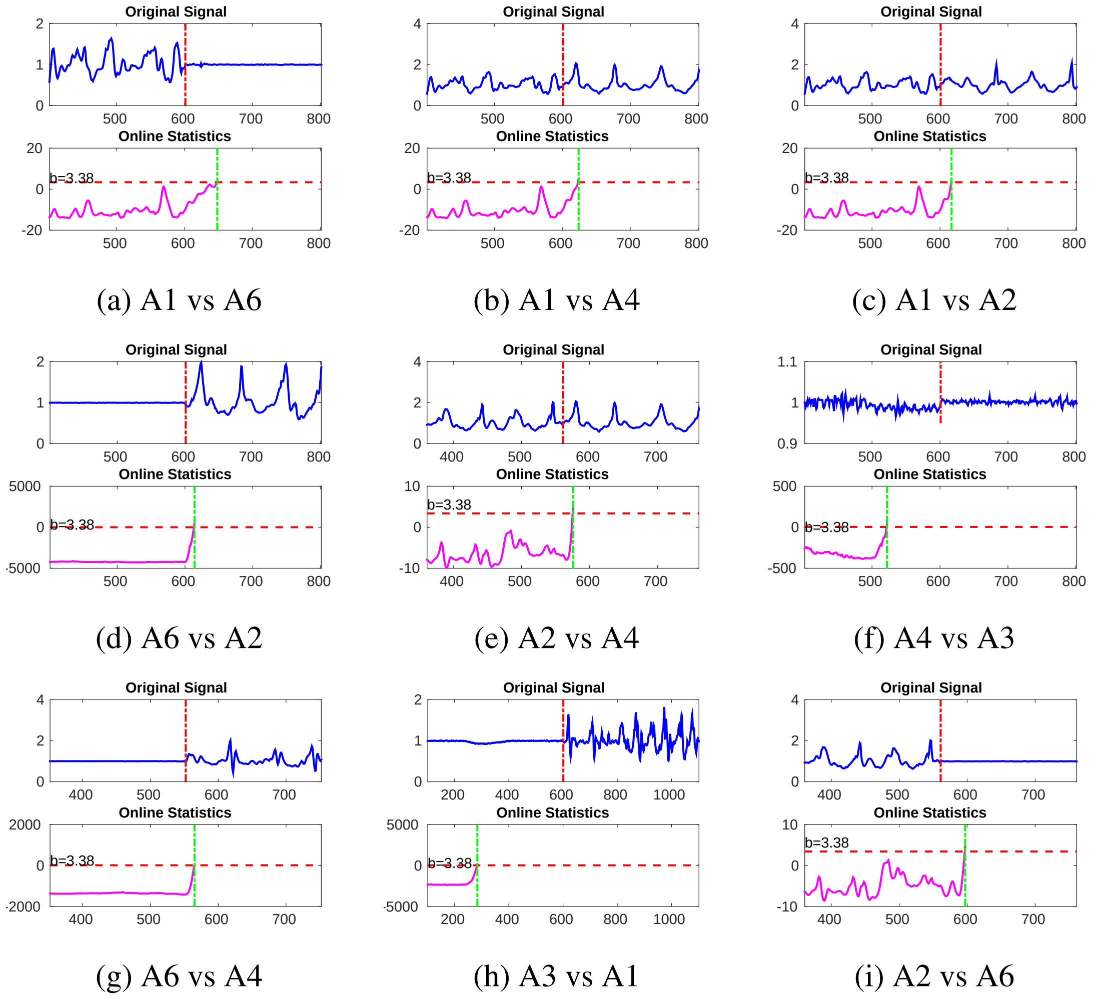
walk/jog, stairUp/stairDown, elevatorUp/elevatorDown,
escalatorUp/escalatorDown, movingWalkway, stay.

We make pairs of signal sequences from different activity categories, and remove the sequences which are too short. We finally get 381 sequences. We tune the parameters using the same way as in CENSREC-1-C experiment. The AUC of our algorithm is **.8871**, compared to **.7161** achieved by baseline algorithm, which greatly improved the performance.

Examples of the signals are shown in Figure A.5. Some sequences are easy to find the change-point, like Figure A.5(a), and A.5(d). Some pairs of the signals are hard to distinguish visually, like Figure A.5(b) and A.5(c). The examples show that our algorithm

[2] http://hasc.jp/hc2011

can tell the change-point from walk to stairUp/stairDown, or from stairUp/stairDown to escalatorUp/escalatorDown. There are some cases when our algorithm raises false alarm. See Figure A.5(h). It finds a change-point during the activity ‘elevatorUp/elevatorDown’. It is reasonable, since this type of action contains the phase from acceleration to uniform motion, and the phase from uniform motion to acceleration.



(a) A1 vs A6

(b) A1 vs A4

(c) A1 vs A2

(d) A6 vs A2

(e) A2 vs A4

(f) A4 vs A3

(g) A6 vs A4

(h) A3 vs A1

(i) A2 vs A6

Figure A.5: Examples of HASC dataset.

APPENDIX B

DETECTING CHANGES IN DYNAMIC EVENTS OVER NETWORKS.

B.1 Proofs of Theorem 1

We show the one dimensional case as an example. The following informal derivation justifies the theorem. Let t be the current time, and let the window-length be L. Recall our notations: $\mathbb{P}$ and $\mathbb{E}$ denote the probability measure and the expectation under the null hypothesis; $\mathbb{P}_{t,\tau,\alpha}$ and $\mathbb{E}_{t,\tau,\alpha}$ denote the probability measure and the expectation under the alternative hypothesis. We also use the notation use $\mathbb{E}[U;A] = \mathbb{E}[U\mathbb{I}\{A\}]$ to denote conditional expectation.

First, to evaluate ARL, we study the probability that the detection statistic exceeds the threshold before a given time m. We will use the change-of-measure technique [128]. Under the null hypothesis, the boundary crossing probability can be written as

$$\begin{aligned}
\mathbb{P}\left[\sup_{t<m,\alpha\in\Theta} \ell_{t,\tau,\alpha} > x\right] &= \mathbb{E}\left[1; \sup_{t<m,\alpha\in\Theta} \ell_{t,\tau,\alpha} > x\right] \\
&= \mathbb{E}\left[\underbrace{\frac{\int_t \int_{\alpha\in\Theta} e^{\ell_{t,\tau,\alpha}} dt d\alpha}{\int_{t'} \int_{\alpha'\in\Theta} e^{\ell_{t',\tau',\alpha'}} dt' d\alpha'}}_{=1}; \sup_{t<m,\alpha\in\Theta} \ell_{t,\tau,\alpha} > x\right] \\
&= \int_t \int_{\alpha\in\Theta} \mathbb{E}\left[\frac{e^{\ell_{t,\tau,\alpha}}}{\int_{t'} \int_{\alpha'\in\Theta} e^{\ell_{t',\tau',\alpha'}} dt' d\alpha'};\right. \\
&\qquad \left.\sup_{t<m,\alpha\in\Theta} \ell_{t,\tau,\alpha} > x\right] dt d\alpha \qquad \text{(B.1)}
\end{aligned}$$

where the last equality follows from changing the order of summation and the expectation.

Using change-of-measure $d\mathbb{P} = e^{-\ell_{t,\tau,\alpha}} d\mathbb{P}_{t,\tau,\alpha}$,, the last equation (B.1) can be written as

$$\int_t \int_{\alpha\in\Theta} \mathbb{E}_{t,\tau,\alpha}\left[\frac{1}{\int_{t'}\int_{\alpha'\in\Theta} e^{\ell_{t',\tau',\alpha'}} dt' d\alpha'}; \right.$$
$$\left. \sup_{t<m,\alpha} \ \ell_{t<m,\tau,\alpha} > x\right] dt d\alpha$$

After rearranging each term and introducing additional notations, the last equation above (B.2) can be written as

$$e^{-x}\int_t \int_{\alpha\in\Theta} \mathbb{E}_{t,\tau,\alpha}\left[\frac{\mathcal{M}_{t,\tau,\alpha}}{\mathcal{S}_{t,\tau,\alpha}} e^{-[\tilde{l}_{t,\tau,\alpha}+m_{t,\tau,\alpha}]}; \right. \\ \left. \tilde{l}_{t,\tau,\alpha} + M_{t,\tau,\alpha} > 0\right] dt d\alpha \tag{B.2}$$

where

$$\mathcal{M}_{t,\tau,\alpha} = \sup_{t'} e^{\ell_{t',\tau',\alpha}-\ell_{t,\tau,\alpha}},$$
$$\mathcal{S}_{t,\tau,\alpha} = \int_{t'} e^{\ell_{t',\tau',\alpha}-\ell_{t,\tau,\alpha}} dt',$$
$$\tilde{l}_{t,\tau,\alpha} = \ell_{t,\tau,\alpha} - x, \quad M_{t,\tau,\alpha} = \log\mathcal{M}_{t,\tau,\alpha}.$$

The final expression is also based on the following approximation. When the interval slightly changes from (τ', t') to (τ, t), α' changes little under the null hypothesis since α' is estimated from data stored in (τ', t'). Therefore, in the small neighborhood of (τ', t'), we may regard α as a constant. This leads to an approximation:

$$\frac{\sup_{t',\alpha'} e^{\ell_{t',\tau',\alpha'}-\ell_{t,\tau,\alpha}}}{\int_{t'}\int_{\alpha'} e^{\ell_{t',\tau',\alpha'}-\ell_{t,\tau,\alpha}} dt' d\alpha'} \approx \frac{\sup_{t'} e^{\ell_{t',\tau',\alpha}-\ell_{t,\tau,\alpha}}}{\int_{t'} e^{\ell_{t',\tau',\alpha}-\ell_{t,\tau,\alpha}} dt'}. \tag{B.3}$$

The representation (B.2) consists of a large deviation exponential decay, given by e^{-x}, and lower order contribution that reside in the expectation. The random variables in expectation are further dissected into random variables that are influenced mainly by local perturbations and the random variable that captures the main part of the variability. We can show that the random variable $\tilde{l}_{t,\tau,\alpha}$, which is referred to as the "global term", has an expectation $(t-\tau)I - x$ under the alternative, and a variance $(t-\tau)\sigma^2$. The other random variables are $\mathcal{M}_{t,\tau,\alpha}$ and $\mathcal{S}_{t,\tau,\alpha}$ and its log $m_{t,\tau,\alpha}$, which are determined by the so-called "local field" $\{\ell_{t',\tau',\alpha} - \ell_{t,\tau,\alpha}\}$ are parameterized by t' when we fix $t-\tau$.

Define $\widehat{\mathcal{M}}_{t,\tau,\alpha}$ and $\widehat{\mathcal{S}}_{t,\tau,\alpha}$ by restricting the integral and maximization only to the range of parameter values that are at most ϵ away from either τ or t. By localization theorem (Theorem 5.2 in [128]), under certain conditions, the local and global components are asymptotically independent, which informs:

$$\begin{aligned} &\mathbb{E}_{t,\tau,\alpha}\left[\frac{\mathcal{M}_{t,\tau,\alpha}}{\mathcal{S}_{t,\tau,\alpha}} e^{-[\tilde{l}_{t,\tau,\alpha}+m_{t,\tau,\alpha}]}; \tilde{l}_{t,\tau,\alpha} + M_{t,\tau,\alpha} > 0\right] \\ &\approx \mathbb{E}_{t,\tau,\alpha}\left[\frac{\widehat{\mathcal{M}}_{t,\tau,\alpha}}{\widehat{\mathcal{S}}_{t,\tau,\alpha}}\right] \frac{1}{\sqrt{(t-\tau)\sigma^2}} \phi\left(\frac{(t-\tau)I - x}{\sqrt{(t-\tau)\sigma^2}}\right). \end{aligned} \tag{B.4}$$

We can further prove (see Appendix B.3) that the expected local rate $\mathbb{E}_{t,\tau,\alpha}\left[\mathcal{M}_{t,\tau,\alpha}/\mathcal{S}_{t,\tau,\alpha}\right]$ only depends on α and is independent of t:

$$\mathbb{E}_{t,\tau,\alpha}\left[\frac{\widehat{\mathcal{M}}_{t,\tau,\alpha}}{\widehat{\mathcal{S}}_{t,\tau,\alpha}}\right] \approx \nu\left(\frac{2\xi}{\eta^2}\right), \tag{B.5}$$

for ξ and η^2 defined in (3.25). The conditions for which these approximations hold are given on Page 56 of [128], and in particular, we need to compute the local rate, which is done in Appendix B.3.

Hence, the probability in (B.2) should be

$$\begin{aligned} &e^{-x} \int_t \int_{\alpha \in (0,1)} \nu\left(\frac{2\xi}{\eta^2}\right) \frac{1}{\sqrt{(t-\tau)\sigma^2}} \phi\left(\frac{(t-\tau)I - x}{\sqrt{(t-\tau)\sigma^2}}\right) d\alpha dt \\ &\approx m e^{-x} \int_{\alpha \in (0,1)} \nu\left(\frac{2\xi}{\eta^2}\right) \frac{1}{\sqrt{(t-\tau)\sigma^2}} \phi\left(\frac{(t-\tau)I - x}{\sqrt{(t-\tau)\sigma^2}}\right) d\alpha. \end{aligned} \tag{B.6}$$

Define C to be the factor that multiplies m in the equation above.

Next, since

$$\mathbb{P}\left[\sup_{t<m,\alpha\in(0,1)} \ell_{t,\tau,\alpha} > x\right] = \mathbb{P}\left[T < m\right],$$

we can relate (B.6) to the ARL $\mathbb{E}[T]$. Note that we can write the tail probability (B.6) in a form $\mathbb{P}\left[T < m\right] = mC[1 + o(1)]$. When $x \to \infty$, from the arguments in [129, 130], we see that the stopping time T is asymptotically exponentially distributed and $\mathbb{P}[T < m] \to 1 - \exp(-Cm)$. As a result, $\mathbb{E}[T] \sim C^{-1}$, which is equivalent to (3.24). Derivations for I, σ^2, ξ and η^2 will be talked about in Appendix B.4.

B.2 First- and second-order statistics of Hawkes processes

We first to characterize the first- and second-order statistics for Hawkes processes, which are useful for evaluating I, σ^2, ξ and η^2. For the defined one-dimensional Hawkes processes and multi-dimensional Hawkes processes, if we choose kernel function $\varphi(t)$ with $\int \varphi(t)dt = 1$, we will have the following two lemmas that are derived from the results in [131]. [132]:

Lemma 15 (First-order statistics for Hawkes processes) *If the influence parameters satisfy $\alpha \in (0, 1)$ (one-dimension) or the spectral norm $\rho(\boldsymbol{A}) < 1$ (high-dimension), then the Hawkes processes are asymptotically stationary and with stationary intensity $m_t = \mathbb{E}_{\mathcal{H}_t}[\boldsymbol{\lambda}_t]$. We further have that for the one-dimensional case*

$$\bar{\lambda} := \lim_{t\to\infty} m_t = \frac{\mu}{1-\alpha}$$

and for the multi-dimensional case

$$\bar{\boldsymbol{\lambda}} := \lim_{t\to\infty} \boldsymbol{m}_t = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}.$$

Lemma 16 (Second-order statistics for Hawkes processes) *For stationary Hawkes pro-*

cesses, the covariance intensity, which is defined as:

$$\boldsymbol{c}(t'-t) = \mathit{Cov}\left[\boldsymbol{\lambda}_t, \boldsymbol{\lambda}_{t'}\right] = \frac{\mathit{Cov}\left[d\boldsymbol{N}_t, d\boldsymbol{N}_{t'}\right]}{dtdt'} \tag{B.7}$$

will only depend on $t'-t$. Then for one-dimensional Hawkes processes, we have:

$$c(\tau) = \begin{cases} \frac{\alpha\beta(2-\alpha)\mu}{2(1-\alpha)^2} e^{-\beta(1-\alpha)\tau}, & \tau > 0; \\ \frac{\mu}{1-\alpha}\delta(\tau), & \tau = 0; \\ c(-\tau), & \tau < 0. \end{cases} \tag{B.8}$$

for the multi-dimensional Hawkes processes

$$\boldsymbol{c}(\boldsymbol{\tau}) = \begin{cases} \beta e^{-\beta(\boldsymbol{I}-\boldsymbol{A})\tau} \boldsymbol{A}\left(\boldsymbol{I} + \frac{1}{2}(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\right) & \\ \quad \cdot \mathit{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right), & \boldsymbol{\tau} > 0; \\ \mathit{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right)\delta(\boldsymbol{\tau}), & \boldsymbol{\tau} = 0; \\ \boldsymbol{c}(-\boldsymbol{\tau})^{\intercal}, & \boldsymbol{\tau} < 0. \end{cases}$$

Proof [Proof of Lemma 15] For multi-dimensional Hawkes processes, by mean field approximation and define $\boldsymbol{m}_t = \mathbb{E}_{\mathcal{H}_t}[\boldsymbol{\lambda}_t]$, we have:

$$\boldsymbol{m}_t = \boldsymbol{\mu} + \boldsymbol{A}\int_{-\infty}^{t} \varphi(t-s)\boldsymbol{m}_s ds \tag{B.9}$$

which can be written as

$$\boldsymbol{m}_t = \left(\boldsymbol{I} + \sum_{n=1}^{\infty} \boldsymbol{A}^n \int_{-\infty}^{t} \varphi^{(\star n)}(s)ds\right)\boldsymbol{\mu}. \tag{B.10}$$

where $\star$ denotes the convolution operation, and $\varphi^{(\star n)}$ denote the n-fold convolution. Let $\Psi(t) = \boldsymbol{A}\varphi(t) + \boldsymbol{A}^2\varphi(t)\star\varphi(t) + \boldsymbol{A}^3\varphi(t)\star\varphi(t)\star\varphi(t) + \cdots = \sum_{n=1}^{\infty}\boldsymbol{A}^n\varphi^{(\star n)}(t)$. And we can write (B.10) as:

$$\boldsymbol{m}_t = \left(\boldsymbol{I} + \int_{-\infty}^{t} \Psi(s)ds\right)\boldsymbol{\mu}.$$

Given a function $f(t)$, we denote its Laplace transform $\mathcal{L}(\cdot)$ as:

$$\widehat{f}(z) = \mathcal{L}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-zt}dt.$$

Next, apply Laplace transform to both sides of equation (B.10). Clearly

$$\widehat{\boldsymbol{m}}(z) = \frac{1}{z}(\boldsymbol{I} - \frac{\beta}{z+\beta}\boldsymbol{A})^{-1}\boldsymbol{\lambda}_0,$$

where

$$\widehat{\Psi}(z) = \sum_{n=1}^{\infty}\left(\frac{\beta}{z+\beta}\right)^n \cdot \boldsymbol{A}^n = (\boldsymbol{I} - \frac{\beta}{z+\beta}\boldsymbol{A})^{-1} - \boldsymbol{I}.$$

By the property of Laplace transformation,

$$\bar{\boldsymbol{\lambda}} := \lim_{t\to\infty}\boldsymbol{m}_t = \lim_{z\to 0} z\widehat{\boldsymbol{m}}(z) = (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}. \tag{B.11}$$

For a special case where $d = 1$, we have $\bar{\lambda} = \mu/(1-\alpha)$. ■

Proof [Proof for Lemma 16] For $\boldsymbol{\tau} > 0$, we have:

$$\begin{aligned}
\boldsymbol{c}(\boldsymbol{\tau}) &= \frac{\mathbb{E}\left[d\boldsymbol{N}_{t+\tau} d\boldsymbol{N}_t^{\intercal}\right]}{(dt)^2} - \bar{\boldsymbol{\lambda}}\bar{\boldsymbol{\lambda}}^{\intercal} = \mathbb{E}\left[\boldsymbol{\lambda}_{t+\tau}\frac{d\boldsymbol{N}_t^{\intercal}}{dt}\right] - \bar{\boldsymbol{\lambda}}\bar{\boldsymbol{\lambda}}^{\intercal} \\
&= \mathbb{E}\left[\left(\boldsymbol{\mu} + \boldsymbol{A}\int_{-\infty}^{t+\tau} \varphi(t+\tau-s) d\boldsymbol{N}_s\right)\frac{d\boldsymbol{N}_t^{\intercal}}{dt}\right] - \bar{\boldsymbol{\lambda}}\bar{\boldsymbol{\lambda}}^{\intercal} \\
&= \boldsymbol{A}\int_{-\infty}^{\tau} \varphi(\tau - s)\boldsymbol{c}(s) ds \\
&= \boldsymbol{A}\varphi(\tau)\text{diag}(\bar{\boldsymbol{\lambda}}) + \boldsymbol{A}\int_{-\infty}^{\tau} \varphi(\tau - s)\boldsymbol{c}(s) ds \\
&= \boldsymbol{A}\varphi(\tau)\text{diag}(\bar{\boldsymbol{\lambda}}) + \boldsymbol{A}\int_{0}^{\infty} \varphi(\tau + s)\boldsymbol{c}(s) ds \\
&\quad + \boldsymbol{A}\int_{0}^{\tau} \varphi(\tau - s)\boldsymbol{c}(s) ds.
\end{aligned}$$

For the last two equalities, we are using the relation, $\boldsymbol{c}(-\tau) = \boldsymbol{c}(\tau)^{\intercal}$ and the fact that when $\tau = 0$ $\boldsymbol{c}(\tau) = \text{diag}(\bar{\boldsymbol{\lambda}})\delta(\tau)$. Note that for Poisson processes, we have $\boldsymbol{c}(\tau) = \text{diag}(\boldsymbol{\lambda})\delta(\tau)$. Now substituting $\varphi(\tau) = \beta e^{-\beta\tau}$ into the above, we have:

$$\begin{aligned}
\boldsymbol{c}(\tau) &= \boldsymbol{A}\beta e^{-\beta\tau}\text{diag}(\bar{\boldsymbol{\lambda}}) + \boldsymbol{A}\int_{0}^{\infty} \beta e^{-\beta(\tau+s)}\boldsymbol{c}(s) ds \\
&\quad + \boldsymbol{A}\int_{0}^{\tau} \beta e^{-\beta(\tau-s)}\boldsymbol{c}(s) ds.
\end{aligned} \tag{B.12}$$

Applying Laplace transform to both sides of (B.12), we obtain

$$\widehat{\boldsymbol{c}}(z) = \frac{\beta}{z+\beta}\boldsymbol{A}\text{diag}(\bar{\boldsymbol{\lambda}}) + \frac{\beta}{z+\beta}\boldsymbol{A}\widehat{\boldsymbol{c}}(\beta) + \frac{\beta}{z+\beta}\boldsymbol{A}\widehat{\boldsymbol{c}}(z),$$

where

$$\begin{aligned}
&\mathcal{L}\left(\int_{0}^{\infty} \beta e^{-\beta(\tau+s)}\boldsymbol{c}(s) ds\right) = \mathcal{L}\left(\beta e^{-\beta\tau}\int_{0}^{\infty} e^{-\beta s}\boldsymbol{c}(s) ds\right) \\
&= \mathcal{L}\left(\beta e^{-\beta\tau}\widehat{\boldsymbol{c}}(\beta)\right) = \frac{\beta}{z+\beta}\widehat{\boldsymbol{c}}(\beta).
\end{aligned} \tag{B.13}$$

Replacing z with β, we obtain

$$\widehat{\boldsymbol{c}}(\beta) = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{A}\text{diag}(\bar{\boldsymbol{\lambda}}).$$

Therefore,

$$\widehat{\boldsymbol{c}}(z) = ((z+\beta)\boldsymbol{I} - \beta\boldsymbol{A})^{-1}\,\beta\boldsymbol{A}\left(\boldsymbol{I} + \frac{1}{2}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{A}\right)$$
$$\cdot\,\text{diag}\left((\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}\right)$$

Using inverse Laplace transform for $\widehat{\boldsymbol{c}}(z)$, we obtain

$$\boldsymbol{c}(\tau) = \mathcal{L}^{-1}\left(\widehat{\boldsymbol{c}}(z)\right) = \beta e^{-\beta(\boldsymbol{I}-\boldsymbol{A})\tau}\boldsymbol{A}\left(\boldsymbol{I} + \frac{1}{2}(\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{A}\right)$$
$$\cdot\,\text{diag}\left((\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\mu}\right), \quad \tau > 0.$$

For a special case $d = 1$, we obtain:

$$c(\tau) = \frac{\alpha\beta(2-\alpha)\mu}{2(1-\alpha)^2}e^{-\beta(1-\alpha)\tau}, \quad \tau > 0.$$

■

B.3 Approximate local rate

To show (B.5), we need to evaluate the mean and variance of the local field $\{\ell_{t+\epsilon,\tau+\epsilon,\alpha} - \ell_{t,\tau,\alpha}\}$ after change-of-measures. From (3.15) we see the the log-likelihood ratio $\ell_{t,\tau,\alpha}$ is an integration from time τ to t. Thus, we can rewrite $\ell_{t+\epsilon,\tau+\epsilon,\alpha}$ into several parts by dissecting the integration region:

$$\int_{\tau+\epsilon}^{t+\epsilon} = \int_{\tau+\epsilon}^{\tau+\epsilon^+} + \int_{\tau+\epsilon^+}^{t+\epsilon^-} + \int_{t+\epsilon^-}^{t+\epsilon}. \tag{B.14}$$

From this we the only overlap of data between $\ell_{t+\epsilon,\tau+\epsilon,\alpha}$ and $\ell_{t,\tau,\alpha}$ is the integration over the interval $(\tau+\epsilon^+, t+\epsilon^-)$. Therefore, we have

$$\begin{aligned}
&\mathbb{E}_{t,\tau,\alpha}[\ell_{t+\epsilon,\tau+\epsilon,\alpha}] = \mathbb{E}\left[\ell_{t+\epsilon,\tau+\epsilon,\alpha} e^{\ell_{t,\tau,\alpha}}\right] \\
&= \mathbb{E}\left[\ell_{\tau+\epsilon^+,\tau+\epsilon,\alpha} e^{\ell_{t,\tau,\alpha}}\right] + \mathbb{E}\left[\ell_{t+\epsilon^-,\tau+\epsilon^+,\alpha} e^{\ell_{t,\tau,\alpha}}\right] \\
&\quad + \mathbb{E}\left[\ell_{t+\epsilon,t+\epsilon^-,\alpha} e^{\ell_{t,\tau,\alpha}}\right] \\
&= \mathbb{E}\left[\ell_{\tau+\epsilon^+,\tau+\epsilon,\alpha}\right] \mathbb{E}\left[e^{\ell_{t,\tau,\alpha}}\right] + \mathbb{E}_{t,\tau,\alpha}\left[\ell_{t+\epsilon^-,\tau+\epsilon^+}\right] \\
&\quad + \mathbb{E}\left[\ell_{t+\epsilon,t+\epsilon^-,\alpha}\right] \mathbb{E}\left[e^{\ell_{t,\tau,\alpha}}\right].
\end{aligned}$$

Due to the property of the likelihood ratio, $\mathbb{E}\left[e^{\ell_{t,\tau,\alpha}}\right] = 1$. Thus, we have:

$$\begin{aligned}
&\mathbb{E}_{t,\tau,\alpha}[\ell_{t+\epsilon,\tau+\epsilon,\alpha}] \\
&= \mathbb{E}\left[\ell_{\tau+\epsilon^+,\tau+\epsilon,\alpha}\right] + \mathbb{E}_{t,\tau,\alpha}\left[\ell_{t+\epsilon^-,\tau+\epsilon^+}\right] + \mathbb{E}\left[\ell_{t+\epsilon,t+\epsilon^-,\alpha}\right] \\
&= -\epsilon^- \frac{\mathbb{E}[\ell_{t,\tau,\alpha}]}{t-\tau} + (t+\epsilon^- - \tau - \epsilon^+) \frac{\mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]}{t-\tau} \\
&\quad + \epsilon^+ \frac{\mathbb{E}[\ell_{t,\tau,\alpha}]}{t-\tau}.
\end{aligned}$$

For the last equality, we use the fact the both $\mathbb{E}[\ell_{t,\tau,\alpha}]$ and $\mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]$ are linear with time interval $(t-\tau)$, which will be proven in Appendix B.4. Finally we have:

$$\begin{aligned}
&\mathbb{E}_{t,\tau,\alpha}[\ell_{t+\epsilon,\tau+\epsilon,\alpha} - \ell_{t,\tau,\alpha}] \\
&= (-\epsilon^- + \epsilon^+) \frac{\mathbb{E}[\ell_{t,\tau,\alpha}]}{t-\tau} - (\epsilon^+ - \epsilon^-) \frac{\mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]}{t-\tau} \\
&= \underbrace{\frac{\mathbb{E}[\ell_{t,\tau,\alpha}] - \mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]}{t-\tau}}_{-\xi<0} |\epsilon|.
\end{aligned}$$

By Jensen's inequality, we can prove that $\mathbb{E}[\ell_{t,\tau,\alpha}] < 0$ and $\mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}] > 0$.

Similarly, we derive the variance of the local field:

$$
\begin{aligned}
& \mathrm{Var}_{\mathrm{t},\tau,\alpha}[\ell_{\mathrm{t}+\epsilon,\tau+\epsilon,\alpha} - \ell_{\mathrm{t},\tau,\alpha}] \\
&= \mathrm{Var}_{\mathrm{t},\tau,\alpha}\left[(\ell_{\tau+\epsilon^+,\tau+\epsilon,\alpha} + \ell_{\mathrm{t}+\epsilon^-,\tau+\epsilon^+,\alpha} + \ell_{\mathrm{t}+\epsilon,\mathrm{t}+\epsilon^-,\alpha}) - \ell_{\mathrm{t},\tau,\alpha}\right] \\
&= \mathrm{Var}_{\mathrm{t},\tau,\alpha}\left[\ell_{\tau+\epsilon^+,\tau+\epsilon,\alpha} - (\ell_{\tau,\tau+\epsilon^+,\alpha} + \ell_{\mathrm{t}+\epsilon^-,\mathrm{t},\alpha}) \quad + \ell_{\mathrm{t}+\epsilon,\mathrm{t}+\epsilon^-,\alpha}\right] \\
&= \mathrm{Var}_{\mathrm{t},\tau,\alpha}\left[\ell_{\tau+\epsilon^+,\tau+\epsilon,\alpha}\right] \\
&\quad + \mathrm{Var}_{\mathrm{t},\tau,\alpha}\left[\ell_{\tau,\tau+\epsilon^+,\alpha} + \ell_{\mathrm{t}+\epsilon^-,\mathrm{t},\alpha}\right] + \mathrm{Var}_{\mathrm{t},\tau,\alpha}\left[\ell_{\mathrm{t}+\epsilon,\mathrm{t}+\epsilon^-,\alpha}\right] \\
&= \mathrm{Var}\left[\ell_{\tau+\epsilon^+,\tau+\epsilon,\alpha}\right] + \mathrm{Var}_{\mathrm{t},\tau,\alpha}\left[\ell_{\tau,\tau+\epsilon^+,\alpha}\right. \\
&\quad \left. + \ell_{t+\epsilon^-,t,\alpha}\right] + \mathrm{Var}\left[\ell_{\mathrm{t}+\epsilon,\mathrm{t}+\epsilon^-,\alpha}\right] \\
&= (\epsilon^+ - \epsilon^-)\frac{\mathrm{Var}[\ell_{\mathrm{t},\tau,\alpha}]}{t-\tau} + (\epsilon^+ - \epsilon^-)\frac{\mathrm{Var}_{\mathrm{t},\tau,\alpha}[\ell_{\mathrm{t},\tau,\alpha}]}{t-\tau} \\
&= \underbrace{\frac{\mathrm{Var}[\ell_{\mathrm{t},\tau,\alpha}] + \mathrm{Var}_{\mathrm{t},\tau,\alpha}[\ell_{\mathrm{t},\tau,\alpha}]}{t-\tau}}_{\eta^2} |\epsilon|.
\end{aligned}
$$

Above, we use the fact that both $\mathrm{Var}[\ell_{t,\tau,\alpha}]$ and $\mathrm{Var}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]$ are approximately linear with time interval $(t-\tau)$, which will be proven in Appendix B.4.

The above derivations show that the asymptotic distribution of $\{\ell_{t+\epsilon,\tau+\epsilon,\alpha} - \ell_{t,\tau,\alpha}\}$, for small $|\epsilon|$ is a two-sided Brownian motion with a negative drift $-\xi$. The variance of an increment of this Brownian motion is η^2. That is, the re-centered process:

$$
\ell_{t+\epsilon,\tau+\epsilon,\alpha} - \ell_{t,\tau,\alpha} = B(\eta^2|\epsilon|) - \xi|\epsilon| \tag{B.15}
$$

with the equality meaning equality in distribution, where B is a two-sided random walk with negative drift. According to Chapter 3 in [128], we obtain (B.5).

B.4 Expectation and variance of log-likelihood ratio under null and alternative distributions

The calculations I, σ^2, ξ and η^2 boil down to evaluating $\mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]$, $\mathrm{Var}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]$, $\mathbb{E}[\ell_{t,\tau,\alpha}]$ and $\mathrm{Var}[\ell_{t,\tau,\alpha}]$, i.e., the expectation and variance of log-likelihood ratio under null and alternative distributions. Below, we will perform the calculation for all likelihoods considered in our paper. The main techniques used are mean-field approximation, Delta method, and Lemma 15 and 16. Below, let $\mathbb{E}_{\mathcal{H}_{t-}}[\cdot]$ denote the conditional expectation for the Hawkes process given the past history.

One-dimension: Poisson to Hawkes.

Assuming stationary and $(t-\tau)$ is large, we can approximate the stationary intensity for the Hawkes process to be $\bar{\lambda}^*$, which is defined as

$$\bar{\lambda}^* = \lim_{t\to\infty} m_t^* = \lim_{t\to\infty} \mathbb{E}_{\mathcal{H}_{t-}}[\lambda_t^*].$$

We use mean field approximation, which assumes each stochastic process λ_t^* has small fluctuations around its mean $\bar{\lambda}^*$: $|\lambda_t^* - \bar{\lambda}^*|/\bar{\lambda}^* \ll 1$. Then we compute the expectation of log-likelihood ratio under alternative hypothesis

$$\begin{aligned}
&\mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}] \\
&= \mathbb{E}_{t,\tau,\alpha}\left[\int_\tau^t \log\left(\lambda_s^*\right) dN_s - \int_\tau^t \log\left(\lambda_s\right) dN_s \right.\\
&\quad \left. - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds\right] \qquad \text{(B.16)}\\
&\approx \mathbb{E}_{\mathcal{H}_{t-}}\left[\int_\tau^t \lambda_s^* \log\left(\lambda_s^*\right) ds \right.\\
&\quad \left. - \int_\tau^t \lambda_s^* \log\left(\lambda_s\right) ds\right] - \int_\tau^t \left(m_s^* - \lambda_s\right) ds. \qquad \text{(B.17)}
\end{aligned}$$

From (B.16) to (B.17), we use the fact that under $\mathbb{P}_{t,\tau,\alpha}$, Ns is a Hawkes random field with conditional intensity λ_s^*. From (B.16) to (B.17), more justifications can be found in [133, 134, 72].

Next, when $(t-\tau)$ is large, we can approximate the stationary intensity for Hawkes process to be $\bar{\lambda}^*$. To approximate $\mathbb{E}_{\mathcal{H}_{t-}}\left[\int_\tau^t \lambda_s^* \log\left(\lambda_s^*\right) ds\right]$, we perform the first order taylor expansion for a new defined function $f(\lambda_s^*) = \lambda_s^* \log\left(\lambda_s^*\right)$ around $\mathbb{E}_{\mathcal{H}_{t-}}[\lambda_s^*] = \bar{\lambda}^*$ (this is based on the Delta method):

$$\lambda_s^* \log\left(\lambda_s^*\right) \approx \bar{\lambda}^* \log\left(\bar{\lambda}^*\right) + \left[\log(\bar{\lambda}^*) + 1\right](\lambda_s^* - \bar{\lambda}^*). \tag{B.18}$$

Taking expectation on both sides of the equation and using $\mathbb{E}_{\mathcal{H}_{t-}}[\lambda_s^*] = \bar{\lambda}^*$, we have

$$\mathbb{E}_{\mathcal{H}_{t-}}\left[\int_\tau^t \lambda^*(s) \log\left(\lambda^*(s)\right) ds\right] \approx \int_\tau^t \bar{\lambda}^* \log(\bar{\lambda}^*) ds.$$

Finally, we have:

$$\begin{aligned} \mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}] &\approx (t-\tau)\left[\bar{\lambda}^* \log\left(\frac{\bar{\lambda}^*}{\mu}\right) - (\bar{\lambda}^* - \mu)\right] \\ &= (t-\tau) \underbrace{\left[\frac{\mu}{1-\alpha} \log\left(\frac{1}{1-\alpha}\right) - \frac{\alpha}{1-\alpha}\mu\right]}_{I}. \end{aligned}$$

On the other hand, under the null distribution and given stationary assumption, we have:

$$\begin{aligned} &\mathbb{E}[\ell_{t,\tau,\alpha}] \\ &= \mathbb{E}\left[\int_\tau^t \log\left(\frac{\lambda_s^*}{\lambda_s}\right) dN_s - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds\right] \\ &\approx \mathbb{E}_{\mathcal{H}_{t-}}\left[\int_\tau^t \lambda_s \log\left(\frac{\lambda_s^*}{\lambda_s}\right) ds - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds\right] \\ &\approx (t-\tau) \underbrace{\left[\mu \log\left(\frac{1}{1-\alpha}\right) - \frac{\alpha}{1-\alpha}\mu\right]}_{I_0}. \end{aligned}$$

For the second equality we use the fact that under $\mathbb{P}$, Ns is a Poisson random field with intensity λ_s. For the last equality, we use mean-field approximation.

Next, we compute the variance of log-likelihood ratio under null distribution and alternative distribution, respectively. Under the alternative distribution,

$$\begin{aligned}&\int_\tau^t \log\left(\frac{\lambda_s^*}{\lambda_s}\right) dN_s - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds \\ \approx & \int_\tau^t \left[\lambda_s^* \log\left(\frac{\lambda_s^*}{\lambda_s}\right) - \lambda_s^*\right] ds + \lambda_s(t-\tau).\end{aligned}$$

Then the only random part is $\int_\tau^t \left[\lambda_s^* \log\left(\frac{\lambda_s^*}{\lambda_s}\right) - \lambda_s^*\right] ds$. Therefore,

$$\mathrm{Var}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}] \approx \mathrm{Var}_{\mathcal{H}_{t-}}\left[\int_\tau^t \left[\lambda_s^* \log\left(\frac{\lambda_s^*}{\lambda_s}\right) - \lambda_s^*\right] ds\right]. \tag{B.19}$$

Again, to use Delta method, we consider a function with respect to λ_s^*:

$$f(\lambda_s^*) = \lambda_s^* \log\left(\frac{\lambda_s^*}{\lambda_s}\right) - \lambda_s^*,$$

and apply the first order taylor expansion around $\mathbb{E}_{\mathcal{H}_{t-}}[\lambda_s^*] = \bar{\lambda}^*$:

$$f(\lambda_s^*) \approx f(\bar{\lambda}^*) + \log\left(\frac{\bar{\lambda}^*}{\lambda_s}\right)\left(\lambda_s^* - \bar{\lambda}^*\right). \tag{B.20}$$

From (B.20), we obtain

$$\begin{aligned}\mathrm{Var}_{\mathcal{H}_{t-}}\left[f(\lambda_s^*)\right] &\approx \mathbb{E}_{\mathcal{H}_{t-}}\left[(f(\lambda_s^*) - f(\lambda^*))^2\right] \\ &\approx \left[\log\left(\frac{\bar{\lambda}^*}{\lambda_s}\right)\right]^2 \mathbb{E}_{\mathcal{H}_t}\left[(\lambda_s^* - \bar{\lambda}^*)^2\right],\end{aligned}$$

where $\mathbb{E}_{\mathcal{H}_{t-}}\left[(\lambda_s^* - \bar{\lambda}^*)^2\right] = \mathrm{Var}_{\mathcal{H}_t}[\lambda_s^*]$. Note that the log-likelihood ratio is an integration from τ to t. When computing the variance, we need to consider $\mathrm{Cov}[\lambda_s^*, \lambda_{s+\tau}^*]$. Under the stationary assumption, from Lemma 16, we obtain an expression for $c(\tau) :=$

$\mathrm{Cov}_{\mathcal{H}_{\mathrm{t-}}}[\lambda^*_{\mathrm{s}}, \lambda^*_{\mathrm{s+\tau}}]$, which only depends on τ. Therefore,

$$\begin{aligned}
&\mathrm{Var}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}] \\
&\approx \left[\log\left(\frac{\bar{\lambda}^*}{\lambda_s}\right)\right]^2 \int_\tau^t \int_\tau^t c(s'-s)dsds' \\
&= \left[\log\left(\frac{\bar{\lambda}^*}{\lambda_s}\right)\right]^2 \left[\int_0^{t-\tau} \lambda^* ds + 2\int_0^{t-\tau}\int_0^s c(v)dvds\right] \\
&= (t-\tau)\left[\log\left(\frac{1}{1-\alpha}\right)\right]^2 \left[\frac{\mu}{1-\alpha} + \frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3} \right.\\
&\quad \left. + \frac{\alpha(2-\alpha)\mu e^{-\beta(1-\alpha)(t-\tau)}}{\beta(1-\alpha)^4(t-\tau)} - \frac{\alpha(2-\alpha)\mu}{\beta(1-\alpha)^4(t-\tau)}\right].
\end{aligned}$$

Moreover, since α is usually a small number, when $(t-\tau)$ is a large number, we may ignore the small terms and further approximate:

$$\begin{aligned}
&\mathrm{Var}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}] \\
&\approx (t-\tau)\underbrace{\left[\log\left(\frac{1}{1-\alpha}\right)\right]^2 \left[\frac{\mu}{1-\alpha} + \frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3}\right]}_{\sigma^2}.
\end{aligned}$$

On the other hand, under the null distribution, we have the variance of the log-likelihood ratio

$$\begin{aligned}
\mathrm{Var}[\ell_{t,\tau,\alpha}] &\approx \left[\log\left(\frac{\bar{\lambda}^*}{\lambda_s}\right)\right]^2 \int_\tau^t \lambda_s ds \\
&= (t-\tau)\underbrace{\mu\left[\log\left(\frac{1}{1-\alpha}\right)\right]^2}_{\sigma_0^2}.
\end{aligned}$$

Multi-dimension: Poisson to Hawkes

The derivations for the multi-dimensional case would follow the same strategy as the one-dimensional case. So we just put the key results here. For the expectation of the log-likelihood ratio under alternative distribution, we have:

$$\begin{aligned}
&\mathbb{E}_{t,\tau,\boldsymbol{A}}[\ell_{t,\tau,\alpha}] \\
&\approx (t-\tau)\left[\bar{\boldsymbol{\lambda}}^{*\intercal}\left(\log(\bar{\boldsymbol{\lambda}}^*)-\log(\boldsymbol{\mu})\right)-\boldsymbol{e}^\intercal(\bar{\boldsymbol{\lambda}}^*-\boldsymbol{\mu})\right] \\
&=(t-\tau)\left[(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\left(\log((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu})-\log(\boldsymbol{\mu})\right)\right. \\
&\quad\left.-\boldsymbol{e}^\intercal((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}-\boldsymbol{\mu})\right].
\end{aligned}$$

where the quantity inside $[\cdot]$ above corresponds to I in this case. Under null, we have

$$\begin{aligned}
\mathbb{E}[\ell_{t,\tau,\alpha}] &\approx (t-\tau)\left[\boldsymbol{\mu}^\intercal\left(\log(\boldsymbol{\lambda}^*)-\log(\boldsymbol{\mu})\right)-\boldsymbol{e}^\intercal(\boldsymbol{\lambda}^*-\boldsymbol{\mu})\right] \\
&=(t-\tau)\left[\boldsymbol{\mu}^\intercal\left(\log((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu})-\log(\boldsymbol{\mu})\right)\right. \\
&\quad\left.-\boldsymbol{e}^\intercal((\boldsymbol{I}-\boldsymbol{A})^{-1}-\boldsymbol{I})\boldsymbol{\mu}\right],
\end{aligned}$$

where the quantity inside $[\cdot]$ above corresponds to I_0 in this case. For the variance of the log-likelihood ratio under alternative, we have

$$\begin{aligned}
&\mathrm{Var}_{t,\tau,\boldsymbol{A}}[\ell_{t,\tau,\boldsymbol{A}}] \\
&=\mathrm{Var}_{t,\tau,\boldsymbol{A}}\left[\sum_{i=1}^d\int_\tau^t\log\left(\frac{\lambda_i(s)}{\mu_i}\right)dN_s^i\right] \\
&=\sum_{i=1}^d\mathrm{Var}_{t,\tau,\boldsymbol{A}}\left[\int_\tau^t\log\left(\frac{\lambda_i(s)}{\mu_i}\right)dN_s^i\right] \\
&+2\sum_{i<j}\mathrm{Cov}_{t,\tau,\boldsymbol{A}}\left[\int_\tau^t\log\left(\frac{\lambda_i(s)}{\mu_i}\right)dN_s^i,\int_\tau^t\log\left(\frac{\lambda_j(s)}{\mu_j}\right)dN_s^j\right].
\end{aligned}\tag{B.21}$$

From Lemma 16, for $s>0$

$$\begin{aligned}
\boldsymbol{c}(s)=\beta e^{-\beta(\boldsymbol{I}-\boldsymbol{A})s}\boldsymbol{A}\left(\boldsymbol{I}+\frac{1}{2}(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\right) \\
\cdot\,\mathrm{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right).
\end{aligned}$$

To compute (B.21), we also need

$$
\begin{aligned}
&\int_\tau^t \int_\tau^t \boldsymbol{c}(s'-s) ds ds' = 2\int_0^{t-\tau}\int_0^s \boldsymbol{c}(v) dv ds \\
=& 2\beta \int_0^{t-\tau}\int_0^s e^{-\beta(\boldsymbol{I}-\boldsymbol{A})v} dv ds \\
&\boldsymbol{A}\left(\boldsymbol{I} + \frac{1}{2}(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\right) \operatorname{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right) \\
=& 2\beta \int_0^{t-\tau} \left(-\frac{1}{\beta}(\boldsymbol{I}-\boldsymbol{A})^{-1}\left(e^{-\beta(\boldsymbol{I}-\boldsymbol{A})s} - \boldsymbol{I}\right)\right) ds \\
&\boldsymbol{A}\left(\boldsymbol{I} + \frac{1}{2}(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\right) \operatorname{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right) \\
=& 2(\boldsymbol{I}-\boldsymbol{A})^{-1} \int_0^{t-\tau} \left(\boldsymbol{I} - e^{-\beta(\boldsymbol{I}-\boldsymbol{A})s}\right) ds \\
&\boldsymbol{A}\left(\boldsymbol{I} + \frac{1}{2}(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\right) \operatorname{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right) \\
\approx& (t-\tau)(\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\left(2\boldsymbol{I} + (\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{A}\right) \\
&\cdot \operatorname{diag}\left((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}\right).
\end{aligned}
$$

Note that when computing $\text{Cov}[dN_s^i, dN_{s'}^i]$, we need to consider an extra term:

$$
\int_\tau^t \int_\tau^t \bar{\boldsymbol{\lambda}}^* \delta(s'-s) ds ds' = \int_0^{t-\tau} \bar{\boldsymbol{\lambda}}^* ds = (t-\tau)\bar{\boldsymbol{\lambda}}^*. \tag{B.22}
$$

After rearranging terms, using the mean-field approximation and Delta method, we obtain

$$
\text{Var}_{t,\tau,\boldsymbol{A}}[\ell_{t,\tau,\boldsymbol{A}}] \approx (t-\tau) \underbrace{\boldsymbol{e}^\intercal (\boldsymbol{H} \circ \boldsymbol{C}) \boldsymbol{e}}_{\sigma^2}, \tag{B.23}
$$

where $\boldsymbol{H}$ and $\boldsymbol{C}$ are defined in Table 3.1.

We compute the variance of the log-likelihood under null distribution. Note that when the data follow Poisson processes, we have $\mathrm{Cov}[N_t^i, N_{t'}^j]_{t\neq t'} = 0$. Therefore,

$$\begin{aligned}
\mathrm{Var}[\ell_{t,\tau,\boldsymbol{A}}] &\approx (t-\tau)\left[\boldsymbol{\mu}^\intercal \left(\log(\bar{\boldsymbol{\lambda}}^*) - \log(\boldsymbol{\mu})\right)^{(2)}\right] && \text{(B.24)}\\
&\approx (t-\tau)\underbrace{\left[\boldsymbol{\mu}^\intercal \left(\log((\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{\mu}) - \log(\boldsymbol{\mu})\right)^{(2)}\right]}_{\sigma_0^2}. && \text{(B.25)}
\end{aligned}$$

One-dimension: Hawkes to Hawkes.

Similarly, we compute the expectation of the log-likelihood ratio under alternative distribution

$$\begin{aligned}
&\mathbb{E}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}]\\
&= \mathbb{E}_{t,\tau,\alpha}\left[\int_\tau^t \log\left(\lambda_s^*\right) dN_s - \int_\tau^t \log\left(\lambda_s\right) dN_s - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds\right]\\
&\approx \mathbb{E}_{\mathcal{H}_{t-}}\left[\int_\tau^t \lambda_s^*\log\left(\lambda_s^*\right) ds - \int_\tau^t \lambda_s^*\log\left(\lambda_s\right) ds - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds\right]\\
&\approx (t-\tau)\left[\bar{\lambda}^*\log(\bar{\lambda}^*) - \bar{\lambda}^*\log(\bar{\lambda}) - (\bar{\lambda}^* - \bar{\lambda})\right]\\
&\approx (t-\tau)\underbrace{\left[\frac{\mu}{1-\alpha^*}\log\left(\frac{1-\alpha}{1-\alpha^*}\right) - \frac{\mu}{1-\alpha^*} + \frac{\mu}{1-\alpha}\right]}_{I},
\end{aligned}$$

where the first approximation is due to that under $\mathbb{P}_{t,\tau,\alpha}$, $N(ds)$ is a Hawkes random field with intensity λ_s^*, and for the latter approximation, we are using mean field approximation and (multivariate) Delta Method given $\mathbb{E}_{\mathcal{H}_{t-}}[\lambda^*(s)] = \bar{\lambda}^*$ and $\mathbb{E}_{\mathcal{H}_{t-}}[\lambda_s] = \bar{\lambda}$. And for the stationary intensity, we have $\bar{\lambda} = \mu/(1-\alpha)$ and $\bar{\lambda}^* = \mu/(1-\alpha^*)$.

Next, the expectation of the log-likelihood ratio under null distribution is given by

$$\begin{aligned}
&\mathbb{E}[\ell_{t,\tau,\alpha}] \\
&= \mathbb{E}\left[\int_\tau^t \log\left(\lambda_s^*\right) dN_s - \int_\tau^t \log\left(\lambda_s\right) dN_s \right.\\
&\quad \left. - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds\right] \\
&\approx \mathbb{E}_{\mathcal{H}_{t-}}\left[\int_\tau^t \lambda_s \log\left(\lambda_s^*\right) ds - \int_\tau^t \lambda_s \log\left(\lambda_s\right) ds \right.\\
&\quad \left. - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds\right] \\
&\approx (t-\tau)\left[\bar{\lambda}\log(\bar{\lambda}^*) - \bar{\lambda}\log(\bar{\lambda}) - (\bar{\lambda}^* - \bar{\lambda})\right] \\
&= (t-\tau)\underbrace{\left[\frac{\mu}{1-\alpha}\log\left(\frac{1-\alpha}{1-\alpha^*}\right) - \frac{\mu}{1-\alpha^*} + \frac{\mu}{1-\alpha}\right]}_{I_0},
\end{aligned}$$

and the variance of the log-likelihood ratio under alternative distribution is given by

$$\begin{aligned}
\ell_{t,\tau,\alpha} &= \int_\tau^t \log\left(\lambda_s^*\right) dN_s - \int_\tau^t \log\left(\lambda_s\right) dN_s \\
&\quad - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds \\
&\approx \int_\tau^t \underbrace{\left[\lambda_s^* \log\left(\lambda_s^*\right) - \lambda_s^* \log\left(\lambda_s\right) - \lambda_s^* + \lambda_s\right]}_{f(\lambda_s^*, \lambda_s)} ds.
\end{aligned}$$

Next, we perform the first order taylor expansion to the newly defined multivariate function with respect to λ_s^* and λ_s:

$$\begin{aligned}
&f(\lambda_s^*, \lambda_s) \\
&\approx f(\bar{\lambda}^*, \bar{\lambda}) + \log\left(\frac{\bar{\lambda}^*}{\bar{\lambda}}\right)\left(\lambda_s^* - \bar{\lambda}^*\right) + \left(1 - \frac{\bar{\lambda}^*}{\bar{\lambda}}\right)(\lambda_s - \bar{\lambda}).
\end{aligned}$$

Based on this, we have

$$\begin{aligned}\text{Var}\left[f(\lambda_s^*, \lambda_s)\right] &= \mathbb{E}\left[\left(f(\lambda_s^*, \lambda_s) - f(\bar{\lambda}^*, \bar{\lambda})\right)^2\right] \\ &\approx \left[\log\left(\frac{\bar{\lambda}^*}{\bar{\lambda}}\right)\right]^2 \text{Var}[\lambda_s^*] + \left(1 - \frac{\bar{\lambda}^*}{\bar{\lambda}}\right)^2 \text{Var}[\lambda_s].\end{aligned}$$

Note that the null intensity λ_s is independent of the alternative intensity λ_s^*. Finally, we have:

$$\begin{aligned}&\text{Var}_{t,\tau,\alpha}[\ell_{t,\tau,\alpha}] \\ &\approx \text{Var}_{t,\tau,\alpha}\left[\int_\tau^t \lambda_s^* \text{log}\left(\lambda_s^*\right) ds - \int_\tau^t \lambda_s^* \text{log}\left(\lambda_s\right) ds \right.\\ &\quad \left. - \int_\tau^t \left(\lambda_s^* - \lambda_s\right) ds\right] \\ &\approx \left[\log\left(\frac{\bar{\lambda}^*}{\bar{\lambda}}\right)\right]^2 \int_\tau^t \int_\tau^t c^*(s' - s) ds ds' \\ &\quad + \left(1 - \frac{\bar{\lambda}^*}{\bar{\lambda}}\right)^2 \int_\tau^t \int_\tau^t c(s' - s) ds ds' \\ &\approx (t - \tau)\left(\left[\log\left(\frac{1-\alpha}{1-\alpha^*}\right)\right]^2 \left[\frac{\mu}{1-\alpha^*} + \frac{\alpha^*(2-\alpha^*)\mu}{(1-\alpha^*)^3}\right]\right. \\ &\quad \left. + \left(1 - \frac{1-\alpha}{1-\alpha^*}\right)^2 \left[\frac{\mu}{1-\alpha} + \frac{\alpha(2-\alpha)\mu}{(1-\alpha)^3}\right]\right).\end{aligned}$$

The factor in the last equation that multiplies $(t - \tau)$ corresponds to σ^2 in this setting. Again, we've ignored some small terms.

Similarly, we can compute the variance of the log-likelihood ratio under null distribution. Under null distribution,

$$\ell_{t,\tau,\alpha} \approx \int_\tau^t \underbrace{\lambda_s \text{log}(\lambda_s^*) - \lambda_s \text{log}(\lambda_s) - \lambda_s^* + \lambda_s}_{f(\lambda_s^*, \lambda_s)} ds. \tag{B.26}$$

Still perform the first order taylor expansion to the new defined function:

$$f(\lambda_s^*, \lambda_s) \approx f(\bar{\lambda}^*, \bar{\lambda}) + \left(\frac{\bar{\lambda}}{\bar{\lambda}^*} - 1\right)(\lambda_s^* - \bar{\lambda}^*) + \log\left(\frac{\bar{\lambda}^*}{\bar{\lambda}}\right)(\lambda_s - \bar{\lambda}).$$

Therefore, using multivariate Delta method

$$\begin{aligned}
&\text{Var}[f(\lambda_s^*, \lambda_s)] \\
&= \mathbb{E}\left[\left(f(\lambda_s^*, \lambda_s) - f(\bar{\lambda}^*, \bar{\lambda})\right)^2\right] \\
&\approx \left(\frac{\bar{\lambda}}{\bar{\lambda}^*} - 1\right)^2 \text{Var}[\lambda_s^*] + \left[\log\left(\frac{\bar{\lambda}^*}{\bar{\lambda}}\right)\right]^2 \text{Var}[\lambda_s].
\end{aligned}$$

Finally we obtain

$$\begin{aligned}
\text{Var}[\ell_{t,\tau,\alpha}] \approx &\left(\frac{\bar{\lambda}}{\bar{\lambda}^*} - 1\right)^2 \int_\tau^t \int_\tau^t c^*(s' - s) ds ds' \\
&+ \left[\log\left(\frac{\bar{\lambda}^*}{\bar{\lambda}}\right)\right]^2 \int_\tau^t \int_\tau^t c(s' - s) ds ds' \\
\approx &(t - \tau)\left(\left[1 - \frac{1 - \alpha^*}{1 - \alpha}\right]^2 \left[\frac{\mu}{1 - \alpha^*} + \frac{\alpha^*(2 - \alpha^*)\mu}{(1 - \alpha^*)^3}\right]\right. \\
&\left. + \left[\log\left(\frac{1 - \alpha}{1 - \alpha^*}\right)\right]^2 \left[\frac{\mu}{1 - \alpha} + \frac{\alpha(2 - \alpha)\mu}{(1 - \alpha)^3}\right]\right).
\end{aligned}$$

The factor in the last equation that multiplies $(t - \tau)$ corresponds to σ_0^2 in this setting.

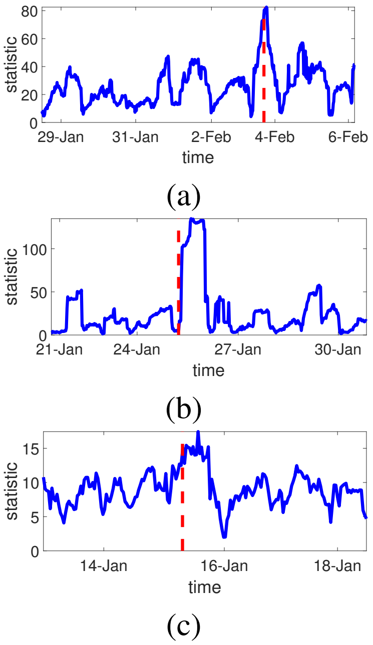
The proof for multi-dimensional case with a transition from the Hawkes process to a Hawkes process is similar and omitted here.

B.5 More real-data examples

The scenario for Fig. 3.9(d) is also interesting as it reflects the activity on the network surrounding Mr. Shkreli, the former chief executive of Turing Pharmaceuticals, who is facing federal securities fraud charges. At Feb. 4th he was invited to congress for a hearing to be questioned about drug price hikes[1].

The fifth example, Fig. 3.9(e) is about Rihanna who announced the release of her new album in a tweet on Jan. 25th. That post was retweeted 170K times and received 280K likes and creates a sudden change in network of her followers.[2]

The last example, in Fig. 3.9(f), demonstrates an increase in the statistic related to the network of Daughter around 25th of January who is attributed to releasing his new album at Jan. 25th.[3]



(a)

(b)

(c)

Figure B.1: Exploratory results on Twitter for the detected change points: (a) Court hearing on Martin Shkreli; (b) Rihanna listens to ANTI; (c) Daughter releases his new album.

[1] http://www.nytimes.com/2016/02/05/business/drug-prices-valeant-martin-shkreli-congress.html

[2] http://jawbreaker.nyc/2016/01/is-rihannas-anti-album-finally-done/

[3] http://www.nme.com/news/daughter/79540

REFERENCES

[1] Z. E. Ross and Y. Ben-Zion, "Automatic picking of direct P, S seismic phases and fault zone head waves," *Geophysical Journal International*, 2014.

[2] Z. Harchaoui, F. Bach, and E. Moulines, "Kernel change-point analysis.," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[3] F. Enikeeva and Z. Harchaoui, "High-dimensional change-point detection with sparse alternatives," *ArXiv:1312.1900*, 2014.

[4] S. Zou, Y. Liang, H. V. Poor, and X. Shi, "Nonparametric detection of anomalous data via kernel mean embedding," *ArXiv:1405.2294*, 2014.

[5] D. Kifer, S. David, and J. Gehrke, "Detecting change in data streams," in *Proceedings of the 30th International Conference on Very Large Data Bases-Volume 30*, VLDB Endowment, 2004, pp. 180–191.

[6] L. Song, Y. Makoto, C. Nigel, and S. Masashi, "Change-point detection in time-series data by direct density-ratio estimation," *Neural Networks*, vol. 43, pp. 72–83, 2013.

[7] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2961–2974, 2005.

[8] W. Zaremba, A. Gretton, and M. Blaschko, "B-test: Low variance kernel two-sample test.," in *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[9] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. Mar, pp. 723–773, 2012.

[10] Z. Harchaoui, F. Bach, O. Cappe, and E. Moulines, "Kernel-based methods for hypothesis testing," *IEEE Signal Processing Magazine*, pp. 87–97, 2013.

[11] B. Yakir, *Extremes in random fields: A theory and its applications*. Wiley, 2013.

[12] D. Siegmund, *Sequential analysis: Tests and confidence intervals*. Springer, 1985.

[13] A. Tartakovsky, I. Nikiforov, and M. Basseville, *Sequential analysis: Hypothesis testing and changepoint detection*. CRC Press, 2014.

[14] L. Gordon and M. Pollak, "An efficient sequential nonparametric scheme for detecting a change of distribution," *Annals of Statistics*, vol. 22, no. 2, pp. 763–804, 1994.

[15] D. Picard, "Testing and estimating change-points in time series," *Advances in applied probability*, vol. 17, no. 04, pp. 841–867, 1985.

[16] E. Brodsky and B. Darkhovsky, *Nonparametric methods in change point problems*. Springer Science & Business Media, 2013, vol. 243.

[17] M. Bibinger, M. Jirak, M. Vetter, *et al.*, "Nonparametric change-point analysis of volatility," *The Annals of Statistics*, vol. 45, no. 4, pp. 1542–1578, 2017.

[18] F. J. . M. Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[19] H. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," *Journal of the American statistical Association*, vol. 62, no. 318, pp. 399–402, 1967.

[20] T. Wang, J. J. Wei, D. Sabatini, and E. Lander, "Genetic screens in human cells using the crispr-cas9 system," *Science*, vol. 343, no. 6166, pp. 80–84, 2014.

[21] G. Fasano and A. Franceschini, "A multidimensional version of the kolmogorov–smirnov test," *Monthly Notices of the Royal Astronomical Society*, vol. 225, no. 1, pp. 155–170, 1987.

[22] M. Csörgő and L. Horváth, "Invariance principles for changepoint problems," *Journal of Multivariate Analysis*, vol. 27, no. 1, pp. 151–168, 1988.

[23] M. Csörgö and L. Horváth, *Limit theorems in change-point analysis*. John Wiley & Sons Inc, 1997, vol. 18.

[24] R. Serfling, *Approximation theorems of mathematical statistics*. John Wiley & Sons, 2009, vol. 162.

[25] H. Dehling, R. Fried, I. Garcia, and M. Wendler, "Change-point detection under dependence based on two-sample u-statistics," in *Asymptotic Laws and Methods in Stochastics*, Springer, 2015, pp. 195–220.

[26] A. Dembo and O. Zeitouni, *Large deviations techniques and applications*, 2nd. Springer, 2009.

[27] Z. Harchaoui and O. Cappé, "Retrospective mutiple change-point estimation with kernels," in *IEEE Workshop on Statistical Signal Processing (SSP)*, IEEE, 2007, pp. 768–772.

[28] S. Arlot, A. Celisse, and Z. Harchaoui, "Kernel change-point detection," *ArXiv preprint arXiv:1202.3878*, vol. 6, 2012.

[29] D. Matteson and N. James, "A nonparametric approach for multiple change point analysis of multivariate data," *Journal of the American Statistical Association*, vol. 109, no. 505, pp. 334–345, 2014.

[30] C. Zou, G. Yin, L. Feng, Z. Wang, *et al.*, "Nonparametric maximum likelihood approach to multiple change-point problems," *The Annals of Statistics*, vol. 42, no. 3, pp. 970–1002, 2014.

[31] B. Scholkopf and A. J. Smola, *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[32] B. Schölkopf, K. Tsuda, and J.-P. Vert, *Kernel methods in computational biology*. Cambridge, MA: MIT Press, 2004.

[33] W. A. Shewhart, *Statistical method from the viewpoint of quality control*. 1939.

[34] D. Maragoni-Simonsen and Y. Xie, "Sequential changepoint approach for online community detection," *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1035–1039, 2015.

[35] A. Ramdas, S. Reddi, B. Póczos, A. Singh, and L. Wasserman, "On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions," in *Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[36] Y. Xie and D. Siegmund, "Sequential multi-sensor change-point detection.," *Annals of Statistics*, vol. 41, no. 2, pp. 670–692, 2013.

[37] P. McCullagh and J. Kolassa, "Cummulants," *Scholarpedia*, vol. 4, no. 3, p. 4699, 2009.

[38] K. Chwialkowski and A. Gretton, "A kernel independence test for random processes," in *International Conference on Machine Learning (ICML)*, 2014.

[39] B. Xie, Y. Liang, and L. Song, "Scale up nonlinear component analysis with doubly stochastic gradients," in *Advances in neural information processing systems (NIPS)*, 2015.

[40] N. Laptev, S. Amizadeh, and I. Flint, “Generic and scalable framework for automated time-series anomaly detection,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1939–1947.

[41] C. Leduc and F. Roueff, “Detection and localization of change-points in high-dimensional network traffic data,” *The Annals of Applied Statistics*, pp. 637–662, 2009.

[42] N. Christakis and J. Fowler, “Social network sensors for early detection of contagious outbreaks,” *PloS one*, vol. 5, no. 9, e12948, 2010.

[43] M. Rodriguez, D. Balduzzi, and B. Schölkopf, “Uncovering the temporal dynamics of diffusion networks,” in *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

[44] S. Myers and J. Leskovec, “The bursty dynamics of the twitter information network,” in *Proceedings of the 23rd International Conference on World Wide Web*, ACM, 2014, pp. 913–924.

[45] R. Ratnam, J. Goense, and M. E. Nelson, “Change-point detection in neuronal spike train activity,” *Neurocomputing*, 2003.

[46] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, “Distinguishing infections on different graph topologies,” *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3100–3120, 2015.

[47] *Twitter statistics*, http://www.internetlivestats.com/twitter-statistics/.

[48] J. Shen and N. Zhang, “Change-point model on nonhomogeneous poisson processes with application in copy number profiling by next-generation dna sequencing,” *The Annals of Applied Statistics*, vol. 6, no. 2, pp. 476–496, 2012.

[49] N. Zhang, B. Yakir, C. Xia, and D. Siegmund, “Scanning a poisson random field for local signals,” *ArXiv preprint arXiv:1406.3258*, 2014.

[50] T. Herberts and U. Jensen, “Optimal detection of a change point in a poisson process for different observation schemes,” *Scandinavian Journal of Statistics*, vol. 31, no. 3, pp. 347–366, 2004.

[51] F. Stimberg, A. Ruttor, M. Opper, and G. Sanguinetti, “Inference in continuous-time change-point models,” in *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[52] J. Pinto, T. Chahed, and E. Altman, "Trend detection in social networks using hawkes processes," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ACM, 2015, pp. 1441–1448.

[53] V. Solo and A. Pasha, "A test for independence between a point process and an analogue signal," *Journal of Time Series Analysis*, vol. 33, no. 5, pp. 824–840, 2012.

[54] M. Farajtabar, N. Du, M. Rodriguez, I. Valera, H. Zha, and L. Song, "Shaping social activity by incentivizing users," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[55] H. Xu, M. Farajtabar, and H. Zha, "Learning granger causality for hawkes processes," *ArXiv preprint arXiv:1602.04511*, 2016.

[56] S. Yang and H. Zha, "Mixture of mutually exciting processes for viral diffusion," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

[57] K. Zhou, H. Zha, and L. Song, "Learning triggering kernels for multi-dimensional hawkes processes," in *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

[58] S. Rajaram, T. Graepel, and R. Herbrich, "Poisson-networks: A model for structured point processes," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, Citeseer, 2005, pp. 277–284.

[59] S. Linderman and R. Adams, "Discovering latent network structure in point process data," *ArXiv preprint arXiv:1402.0914*, 2014.

[60] E. Hall and R. Willett, "Tracking dynamic point processes on networks," *ArXiv preprint arXiv:1409.0031*, 2014.

[61] M. Farajtabar, Y. Wang, M. Rodriguez, S. Li, H. Zha, and L. Song, "Coevolve: A joint point process model for information diffusion and network co-evolution," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[62] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, 1993.

[63] A. Tartakovsky, I. Nikiforov, and M. Basseville, *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. Chapman and Hall/CRC, 2014.

[64] A. W. Shewhart, “Economic control of quality of manufactured product,” *Preprinted by ASQC quality press*, 1931.

[65] E. S. Page, “Continuous inspection schemes,” *Biometrika*, vol. 41, no. 1/2, pp. 100–115, 1954.

[66] ——, “A test for a change in a parameter occurring at an unknown point,” *Biometrika*, vol. 42, no. 3/4, pp. 523–527, 1955.

[67] W. A. Shiryaev, “On optimal methods in quickest detection problems,” *Theory Prob. Appl.*, vol. 8, pp. 22 –46, 1963.

[68] S. W. Roberts, “A comparison of some control chart procedures,” *Technometrics*, no. 8, pp. 411–430, 1966.

[69] T. L. Lai, “Sequential changepoint detection in quality control and dynamical systems,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 613–658, 1995.

[70] A. Ihler, J. Hutchins, and P. Smyth, “Adaptive event detection with time-varying poisson processes,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, 2006.

[71] Y. Mei, S. Han, and K. Tsui, “Early detection of a change in poisson rate after accounting for population size effects,” *Statistica Sinica*, pp. 597–624, 2011.

[72] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes: General theory and structure*. Springer Science & Business Media, 2007.

[73] J. G. Ransmussen, *Temporal point processes: The conditional intensity function*, Lecture Notes, 2011.

[74] N. Barbieri, F. Bonchi, and G. Manco, “Influence-based network-oblivious community detection,” in *IEEE 13th Int. Conf. on Data Mining (ICDM)*, 2013.

[75] D. Siegmund, *Sequential Analysis: Test and Confidence Intervals*. Springer, 1985.

[76] A. Tartakovsky, I. Nikiforov, and M. Basseville, *Sequential analysis: Hypothesis Testing and Changepoint Detection*. Chapman and Hall/CRC, 2014.

[77] A. Simma and M. Jordan, “Modeling events with cascades of poisson processes,” in *Association for Uncertainty in Artificial Intelligence (UAI)*, 2012.

[78] G. Casella and C. Robert, *Monte Carlo Statistical Methods*. Springer, 2004.

[79] G. Casella and R. L. Berger, *Statistical inference*. Cengage Learning, 2001.

[80] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2009, pp. 497–506.

[81] M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha, "Multistage campaigning in social networks," *ArXiv preprint arXiv:1606.03816*, 2016.

[82] M. Gomez Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2010, pp. 1019–1028.

[83] M. Farajtabar, Y. Wang, M. G. Rodriguez, S. Li, H. Zha, and L. Song, "Coevolve: A joint point process model for information diffusion and network co-evolution," in *Advances in Neural Information Processing Systems*, 2015, pp. 1954–1962.

[84] N. Du, L. Song, M. G. Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *Advances in neural information processing systems*, 2013, pp. 3147–3155.

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

[86] H. Mei and J. M. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," in *Advances in Neural Information Processing Systems*, 2017, pp. 6754–6764.

[87] S. Xiao, M. Farajtabar, X. Ye, J. Yan, X. Yang, L. Song, and H. Zha, "Wasserstein learning of deep generative point process models," in *Advances in Neural Information Processing Systems*, 2017, pp. 3250–3259.

[88] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems*, 2015, pp. 2980–2988.

[89] J. Grandell, *Doubly stochastic Poisson processes*. Springer, 2006, vol. 529.

[90] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 1. MIT press Cambridge, 1998, vol. 1.

[91] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, ACM, 2004, p. 1.

[92] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, 2000, pp. 663–670.

[93] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning.," in *AAAI*, Chicago, IL, USA, vol. 8, 2008, pp. 1433–1438.

[94] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 4565–4573.

[95] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in neural information processing systems*, 2007, pp. 513–520.

[96] B. Kim and J. Pineau, "Maximum mean discrepancy imitation learning.," in *Robotics: Science and systems*, 2013.

[97] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," *ArXiv preprint arXiv:1505.03906*, 2015.

[98] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[99] A. Smola, A. Gretton, L. Song, and B. Schölkopf, "A hilbert space embedding for distributions," in *International Conference on Algorithmic Learning Theory*, Springer, 2007, pp. 13–31.

[100] B. Sriperumbudur, K. Fukumizu, and G. Lanckriet, "On the relation between universality, characteristic kernels and rkhs embedding of measures," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 773–780.

[101] M. G. Genton, "Classes of kernels for machine learning: A statistics perspective," *Journal of machine learning research*, vol. 2, no. Dec, pp. 299–312, 2001.

[102] R. A. Adams and J. J. Fournier, *Sobolev spaces*. Academic press, 2003, vol. 140.

[103] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv preprint arXiv:1412.6980*, 2014.

[104] J. F. C. Kingman, *Poisson processes*. Wiley Online Library, 1993.

[105] T. Omi, Y. Hirata, and K. Aihara, “Hawkes process model with a time-dependent background rate and its application to high-frequency financial data,” *Physical Review E*, vol. 96, no. 1, p. 012 303, 2017.

[106] J. Pfanzagl, *Parametric statistical theory*. Walter de Gruyter, 2011.

[107] P. Reynaud-Bouret, S. Schbath, *et al.*, “Adaptive estimation for hawkes processes; application to genome analysis,” *The Annals of Statistics*, vol. 38, no. 5, pp. 2781–2822, 2010.

[108] E. Bacry, I. Mastromatteo, and J.-F. Muzy, “Hawkes processes in finance,” *Market Microstructure and Liquidity*, vol. 1, no. 01, p. 1 550 005, 2015.

[109] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, “Seismic: A self-exciting point process model for predicting tweet popularity,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2015, pp. 1513–1522.

[110] M. Farajtabar, N. Du, M. G. Rodriguez, I. Valera, H. Zha, and L. Song, “Shaping social activity by incentivizing users,” in *Advances in neural information processing systems*, 2014, pp. 2474–2482.

[111] S. Li, S. Xiao, S. Zhu, N. Du, Y. Xie, and L. Song, “Learning temporal point processes via reinforcement learning,” in *Advances in Neural Information Processing Systems*, 2018, pp. 10 781–10 791.

[112] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” *ArXiv preprint arXiv:1702.08608*, 2017.

[113] C. H. Lee and H.-J. Yoon, “Medical big data: Promise and challenges,” *Kidney research and clinical practice*, vol. 36, no. 1, p. 3, 2017.

[114] R. R. Smullyan, *First-order logic*. Springer Science & Business Media, 2012, vol. 43.

[115] Y. Ogata, “Statistical models for earthquake occurrences and residual analysis for point processes,” *Journal of the American Statistical association*, vol. 83, no. 401, pp. 9–27, 1988.

[116] Y Ogata and D Vere-Jones, “Inference for earthquake models: A self-correcting model,” *Stochastic processes and their applications*, vol. 17, no. 2, pp. 337–347, 1984.

[117] M. Achab, E. Bacry, S. Gaïffas, I. Mastromatteo, and J.-F. Muzy, “Uncovering causality from multivariate hawkes integrated cumulants,” *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6998–7025, 2017.

[118] J. F. Allen, “Maintaining knowledge about temporal intervals,” in *Readings in qualitative reasoning about physical systems*, Elsevier, 1990, pp. 361–372.

[119] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, “Mimic-iii, a freely accessible critical care database,” *Scientific data*, vol. 3, p. 160 035, 2016.

[120] F Kanoun, Z. A. Ben, B Zouari, and F. K. Ben, “Insulin therapy may increase blood pressure levels in type 2 diabetes mellitus.,” *Diabetes & metabolism*, vol. 27, no. 6, pp. 695–700, 2001.

[121] L. M. Frydrych, F. Fattahi, K. He, P. A. Ward, and M. J. Delano, “Diabetes and sepsis: Risk, recurrence, and ruination,” *Frontiers in endocrinology*, vol. 8, p. 271, 2017.

[122] R. D. Lopes, J. D. Horowitz, D. A. Garcia, M. A. Crowther, and E. M. Hylek, “Warfarin and acetaminophen interaction: A summary of the evidence and biologic plausibility,” *Blood*, vol. 118, no. 24, pp. 6269–6273, 2011.

[123] D. Siegmund, B. Yakir, and N. Zhang, “Tail approximations for maxima of random fields by likelihood ratio transformations,” *Sequential Analysis*, vol. 29, no. 3, pp. 245–262, 2010.

[124] D. Siegmund and E. S. Venkatraman, “Using the generalized likelihood ratio statistic for sequential detection of a change-point,” *Annals of Statistics*, no. 23, pp. 255–271, 1995.

[125] D. Siegmund and B. Yakir, “Detecting the emergence of a signal in a noisy image,” *Statistics and Its Inference*, no. 1, pp. 3–12, 2008.

[126] B. Yakir, “Multi-channel change-point detection statistic with applications in dna copy-number variation and sequential monitoring,” in *Proceedings of Second International Workshop in Sequential Methodologies*, 2009, pp. 15–17.

[127] R. Arratia, L. Goldstein, and L. Gordon, “Two moments suffice for Poisson approximations: The Chen-Stein method,” *The Annals of Probability*, pp. 9–25, 1989.

[128] B. Yakir, *Extremes in random fields: A theory and its applications*. John Wiley & Sons, 2013.

[129] D. Siegmund and E. S. Venkatraman, “Using the generalized likelihood ratio statistic for sequential detection of a change-point,” *Ann. Statist.*, vol. 23, no. 1, pp. 255 –271, 1995.

[130] D. O. Siegmund and B. Yakir, “Detecting the emergence of a signal in a noisy image,” *Statistics and Its Inference*, vol. 1, pp. 3–12, 2008.

[131] A. G. Hawkes, “Spectra of some self-exciting and mutually exciting point processes,” *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[132] E. Bacry and J. Muzy, “Second order statistics characterization of hawkes processes and non-parametric estimation,” *ArXiv preprint arXiv:1401.0903*, 2014.

[133] D. Daley and D. Jones, “Scoring probability forecasts for point processes: The entropy score and information gain,” *Journal of Applied Probability*, pp. 297–312, 2004.

[134] D. Jones, “Probabilities and information gain for earthquake forecasting,” *Selected Papers From Volume 30 of Vychislitel’naya Seysmologiya*, pp. 104–114, 1998.

VITA

Shuang Li was born in Puyang, Henan Province, China. She received B.E. in Automation from University of Science and Technology, China in 2011, and M.S. in Statistics from Georgia Institute of Technology in 2014. She joined the Ph.D. program in the School of Industrial and Systems Engineering at Georgia Institute of Technology after that. She will obtain her Ph.D. in Industrial Engineering (specialization in Statistics) in August, 2019.