Randomized Fourier Transformations for Spatio-Temporal Hawkes Processes

Fatih Ilhan, Student, IEEE, and Suleyman S. Kozat, Senior Member, IEEE

Abstract—We investigate spatio-temporal network analysis using point processes, in particular, spatio-temporal Hawkes processes. To learn the parameters related to the process, we introduce a novel optimization method based on random projections and gradient descent. More specifically, we replace the spatial kernel calculations with vector products by applying randomized Fourier feature-based transformations. In this way, the system described by the process is represented within closed-form in terms of matrix notations, which increases the scalability considerably compared to the explicit calculations of kernel outputs. During optimization, we perform gradient descent, which properly handles some positivity and orthogonality constraints to learn the internal parameters of the process. The experiment results show the improvements achieved by the proposed method with respect to the conventional inference methods in the spatio-temporal Hawkes process literature. We demonstrate the performance of our method in terms of fitting capability in synthetic and real-world datasets and perform network analysis through the interpretation of learned parameters.

Index Terms—spatio-temporal, Hawkes process, random Fourier features, gradient descent, network analysis

I. INTRODUCTION

A. Preliminaries

We study spatio-temporal network analysis using point processes, which has several applications in signal processing, computer networks, security and forecasting applications [1]–[6]. Most of the real-world events exhibit certain spatio-temporal patterns such as correlation, causation, and excitation, which can be modeled as a network whose latent structure is reflected into real-world with their realizations. Successfully modeling and learning this structure carries importance due to its promising applications such as network analysis, event prediction and hotspot detection [7]–[11]. In particular, spatio-temporal network analysis is an effective tool to obtain insights from complex spatio-temporal data.

To capture the dynamics of the underlying system which generates the event sequence, using regression-based methods might seem a natural approach. This could be achieved by constructing a three-dimensional grid by dividing the space into spatio-temporal regions and aggregating the events inside each region. Depending on the task, the number of events in these regions and their relations with spatiotemporally neighboring regions can be analyzed. However, this approach has several disadvantages, such as sparsity caused by the non-homogenous distribution of events when they are rare in some regions or time intervals [12]. Regression-based models have another significant issue, which is the Modifiable Areal Unit Problem, i.e., these models are more likely to have high variance in terms of estimated regression coefficients depending on the choice of region boundaries [13].

To remedy these problems, point processes are used to capture the dynamics of the event sequence by expressing their rate of occurrences with an intensity function conditioned on the history [14]. In our problem, events are described by their locations, times and types. Therefore, we consider a multi-dimensional form of point processes called as spatio-temporal point processes as done in the previous works [9], [10], [12]. This approach has been applied to various real-world scenarios such as seismological modeling of earthquakes and aftershocks [11], [15], criminological modeling of the dynamics of crime incidents [4], [8], forecasting of disease outbreaks [12], social network analysis [2], [10], and many others. When carefully analyzed, the behavior of underlying systems can vary among different contexts. To this end, several forms of point processes have been proposed with different characteristics, such as Poisson processes, self-correcting processes [16], and self-exciting processes [14]. In this study, we consider spatio-temporal Hawkes process that has a self-exciting nature by its default form in which the intensity value is triggered by past events. This modeling was first applied for earthquake prediction [11] and then successfully adapted to other applications such as crime analysis [4], [8].

B. Prior Art and Comparisons

In order to learn the parameters of a spatio-temporal Hawkes process, there exist several optimization methods in the point process literature, most notably, expectation-maximization (EM) [10] algorithm and stochastic declustering [17]. However, EM algorithm can suffer from instability due to bad initialization and slow convergence in regions, where the likelihood function is flat [18], [19], and stochastic declustering is a specially designed method for earthquake analysis [17]. Recently, gradient descent-based optimization methods has been proposed in the context of temporal point processes [3], [20]–[22]. Gradient-descent based methods are shown to be simple yet effective, particularly for neural networks, thus they have increasingly been favored in signal processing, machine learning, and computer vision literature. Nevertheless, employing gradient descent in the spatio-temporal case is not as straightforward as in the temporal case.

The difficulty of applying gradient descent to the problem of learning spatio-temporal Hawkes processes lies within the structure of the likelihood function. Since the likelihood function includes multi-dimensional integrals of kernel outputs, maximum likelihood estimation with gradient-based methods
is not viable directly. The conditional intensity function of a temporal point process is only defined along the temporal dimension, hence expressing likelihood objective in a differentiable manner, and applying gradient descent-based optimization is rather straightforward compared to the spatio-temporal case. Some previous works also propose to employ Monte-Carlo estimation if there are intractable terms in the likelihood function or parameter gradients [23]. However, relying on Monte-Carlo estimation is also problematic due to the increased space after introducing a spatial dimension, which may result in degraded approximation performance [24]. To handle these issues, we utilize random Fourier features [25], which approximate the output of a shift-invariant continuous kernel using the inner products of the embedded vectors. In the context of this work, we replace the pairwise kernel calculations with randomized vector products, which eases us to derive the likelihood and its derivatives analytically. This randomized transformation method has already been used successfully to increase the scalability of kernel-based methods such as support vector machines (SVMs) [26]–[30]. From the neural network perspective, it can also be interpreted as a perceptron layer with randomly initialized weights and sinusoidal activation function, where the weights are not being updated and they are sampled from a distribution related to the spectral distribution of the kernel. In the machine learning literature, this architecture is called as extreme learning machines (ELM) [31] that have universal approximation capability as the number of nodes (embedding dimensions) goes to infinity [25], [32].

Since we only have to approximate the kernel outputs of two-dimensional spatial vectors, low embedding dimensions suffice with negligible approximation errors. This enables us to replace complicated pairwise kernel calculations with more efficient and scalable matrix operations. To handle the numerical constraints over the process parameters, we also utilize reparameterization techniques and projected gradient-descent.

C. Contributions

Our main contributions are as follows:

1) We introduce a gradient-based optimization method to learn the parameters of a spatio-temporal Hawkes process through maximizing the likelihood.

2) We utilize randomized Fourier transformations to replace kernel operations with vector products, and express the likelihood function in terms of matrix operations.

3) We demonstrate the fitting performance of the proposed method over simulated and real-world datasets.

4) We perform network analysis over spatio-temporal event sequences through the interpretation of learned parameters.

D. Organization

The remainder of the paper is organized as follows. We give the form of the spatio-temporal Hawkes process and introduce the optimization problem in Section II. Then we provide the matrix formulations to express the likelihood function in closed-form using random Fourier features in Section III-B. Then, we analytically obtain the derivatives of the likelihood with respect to process parameters in Section III-C. In Section III-D, we give the gradient-based optimization algorithm for maximum likelihood estimation under parameter constraints. We analyze the fitting performance of the proposed method over simulated and real-world datasets and perform network analysis in Section IV. We conclude the paper in Section V with several remarks.

II. Problem Description

In this paper, we use the numerator layout notation. All vectors are column vectors and denoted by boldface lower case letters. Matrices are denoted by boldface upper case letters. For any vector $x$ or matrix $X$, $x^T$ and $X^T$ are the corresponding transposes. $\|x\|$ is the $\ell^2$-norm of $x$.

We study spatio-temporal point processes, where we observe an event sequence $E = \{e_i\}_{i=1}^N$, where $N$ is the total number of events and $e_i = \{u_i, t_i, s_i\}_{i=1}^N$ is the $i$th event in the sequence with type $u_i \in \mathbb{N}$, time $t_i \in \mathbb{R}$ and location $s_i = [x_i, y_i]^T \in \mathbb{R}^2$. The goal is to model this sequence as a point process and estimate its underlying parameters that are expressed as $\theta$.

Although there are several types of point process models proposed in the literature [12], we particularly consider Hawkes process due to the motivations explained in Section I. We model the dynamics of the point process $E$ in the form of a spatio-temporal Hawkes process and solve the underlying optimization problem with the aid of random Fourier features and gradient descent.

A. Temporal Point Processes

A temporal point process is a stochastic process that consists of realizations of subsequent events in discrete time $t_i \in \mathbb{R}$ with $i \in \mathbb{Z}$. We can interpret a temporal point process by specifying the distribution of the time distance between subsequent events (inter-event times). Let $f^*(t) = f(t|\mathcal{H}_t)$ be the conditional density function for the time of the next event given the time history of events $\mathcal{H}_t$. To express the past dependence in an evolutionary point process, conditional intensity function is defined as follows [12], [14],

$$\lambda^*(t) = \frac{f^*(t)}{1 - F^*(t)}.$$  \hspace{1cm} (1)

The conditional intensity function is useful to describe a temporal point process as a counting process as well [14], which can be expressed as

$$\lambda^*(t)dt = \mathbb{E}(N(dt)|\mathcal{H}_t),$$  \hspace{1cm} (2)

where $dt$ is an infinitesimal time interval around $t$, and $N(dt)$ is the counting random variable for the number of events occurred in $dt$. It is important to note that $\lambda^*(t) \geq 0$ by definition.

We can express the conditional density function $f^*(t)$ in terms of conditional intensity function $\lambda^*(t)$ using (1) as

$$f^*(t) = \lambda^*(t)e^{-\int_{t_n}^t \lambda^*(\tau)d\tau},$$ \hspace{1cm} (3)

$$\Lambda_{\lambda^*}(t) = \int_{t_n}^t \lambda^*(\tau)d\tau,$$ \hspace{1cm} (4)
where \( t_n \) is the time of the last event before \( t \). Here, the conditional intensity function can have many forms. As a simple example, in the case of a Poisson process, \( \lambda'(t) = \lambda(t) = \lambda \), i.e. value of the conditional intensity function is constant through time.

### B. Hawkes Processes

Unlike the Poisson Process, Hawkes process has an evolutionary nature, in which the events excite each other depending on their types and distance as expressed in the following form:

\[
\lambda^*_u(t) = \mu_u + \sum_{j,t_j < t} K_{u_j u} g(t, t_j),
\]

where \( \mu_u \) denotes the background conditional intensity, \( K_{u_j u} \) is the excitation of event type \( u_j \) over \( u \) for triggering conditional intensity and \( g(t, t_j) \) is the output of the temporal triggering evaluated at event times \( t \) and \( t_j \). This form enables to model the point processes that show temporally clustered patterns.

### C. Spatio-Temporal Hawkes Processes

In the spatio-temporal case, each event also has a spatial vector \( (s) \) that describes its location. While expressing the conditional intensity function, we consider the following form in our problem\(^1\):

\[
\lambda_u(t, s) = \mu_u(s) + \gamma_u(t, s),
\]

where \( \mu_u(s) \) denotes the base conditional intensity for spatial vector \( s \) and event type \( u \), and \( \gamma_u(t, s) \) denotes the triggering conditional intensity for any time \( t \), \( s \) and \( u \). We can parametrize the base and triggering conditional intensities in (6) as follows,

\[
\mu_u(s) = \frac{1}{T} \sum_{j=1}^{N} K_{ij} u_j g_2(s, s_j),
\]

\[
\gamma_u(t, s) = \sum_{j, t_j < t} K_{ij} g_1(t, t_j) g_2(s, s_j),
\]

where \( g_1 \) and \( g_2 \) are the temporal and spatial kernel functions. These functions can be expressed as

\[
g_1(t, t_j) = w_{u_j u} e^{-w_{u_j u}(t - t_j)}
\]

and

\[
g_2^{(1)}(s, s_j) = \frac{1}{2\pi} |\Sigma^{(1)}|^{-1/2} e^{-\frac{1}{2}(s - s_j)^T \Sigma^{(1)^{-1}} (s - s_j)},
\]

where \( T \) is the total duration of the event sequence, \( N \) is the total number of events, \( \Sigma^{(1)} \) is the covariance matrix of spatial Gaussian kernel for intensity component \( (.) \), and \( w_{u_j u} \geq 0 \) is the decay rate of the intensity triggered by event type \( u_j \) over \( u \).

The excitation values \( (K_{ij}) \) and weight decays \( (w_{ij}) \) are expressed in form of matrices \( K \) and \( W \) where \( K_{ij}, w_{ij} \geq 0 \). The covariance matrix contains the following terms:

\[
\Sigma = \begin{bmatrix}
\sigma_x^2 & \rho \sigma_x \sigma_y \\
\rho \sigma_x \sigma_y & \sigma_y^2
\end{bmatrix},
\]

where \( \sigma_x \) and \( \sigma_y \) are the standard deviations of the spatial kernel over \( x \) and \( y \) axes, and \( \rho \) is the correlation parameter between \( x \) and \( y \) axes. The multivariate normal distribution is said to be non-degenerate when the symmetric covariance matrix \( \Sigma \) is positive definite. In this case, \( g_2(s, s_j) \) will have an invertible covariance matrix and density.

---

\(^1\)Note that \(^*\) sign, which denotes the conditionality on history will be dropped from now on for the sake of notational simplicity.

---

Fig. 1: Line plots of conditional intensity values in time for four different locations \( (s = \{(1, 1), (-2, 1), (-2, -1), (1, -1)\}) \), and heat maps of conditional intensity values in space for four different time frames \( (t = \{1, 4, 6, 9\}) \). Each red dot represents an event.

To provide an intuitive explanation, in Fig. 1, we give a representative graph that visualizes the conditional intensity of certain points along space and time. Each event realization has a triggering effect which decays in space with Gaussian kernel and in time with exponential kernel.

It is still possible to use the form in (3) to express the conditional density function for the spatio-temporal case as

\[
f_u(t, s) = \lambda_u(t, s) \exp(-\Lambda_\lambda(t)),
\]

where

\[
\Lambda_\lambda(t) = \sum_{u=1}^{U} \int_{t_n}^{t} \int_{s' \in S} \lambda_u(t', s') ds' dt'.
\]

To estimate the optimum parameter set \( \theta = \{K^{(\mu)}, K^{(\gamma)}, W, \Sigma^{(\mu)}, \Sigma^{(\gamma)}\} \), we follow maximum likelihood estimation approach. To this end, the negative log-likelihood over the real event sequence \( \mathcal{E} = \{e_i\}_{i=1}^{N} \) is minimized where \( N \) denotes the number of events. The objective is given below:

\[
\hat{\theta} = \arg \min_{\theta} \mathcal{L},
\]

where \( \mathcal{L} \) is the negative log-likelihood and can be expressed as

\[
\mathcal{L} = -\log \left( \prod_{i=1}^{N} f_u(t_i, s_i) \right)
\]

\[
= -\sum_{i=1}^{N} \log \lambda_u(t_i, s_i) + \sum_{i=1}^{N} \Lambda_\lambda(t_i),
\]
where the second term involving $\Lambda_k(t_i)$ values can be interpreted as a regularizer which prevents to produce high intensity values over all space defined by $T$ and $S$.

We point out that some parameters included in $\theta$ are reached indirectly through intermediate parameters during optimization to handle numerical constraints such as positivity of $K^{(1)}$ and $W$, and unique properties of covariance matrices. Methods to handle these constraints during optimization are explained in Sections III-D.

**D. Random Fourier Features**

Random Fourier features provides an efficient way to approximate the output of a shift-invariant continuous kernel $k(x, y)$ with $x, y \in \mathbb{R}^d$ [25]. This technique embeds kernel inputs $(x, y)$ into a $D$-dimensional Euclidean inner product space using a transformation matrix $F \in \mathbb{R}^{D \times D}$ and approximates $k(x, y)$ through the inner product of embedded vectors. Although it is widely used to scale up kernel based methods such as SVM for large datasets, [29], [30] we use it to replace complex kernel calculations with straightforward methods such as SVM for large datasets. The main idea behind random Fourier features is to approximate the kernel function by its Monte-Carlo estimate

$$k(x, y) = \int_{\mathbb{R}^d} p(w) e^{i w^T (x-y)} dw = E_{\omega} [\zeta_{\omega}(x) \zeta_{\omega}(y)^*],$$

where $\zeta_{\omega}(x) = e^{i w^T x}$, and $e^*$ denotes the complex conjugate of $c \in C$. The main idea behind random Fourier features is to approximate the kernel function by its Monte-Carlo estimate as follows,

$$\tilde{k}(x, y) = \frac{1}{D} \sum_{i=1}^{D} z_i^T(x) z_i(y),$$

where $z_i(x) = \sqrt{2} \cos(w_i^T x + b)$ with $w \sim p(w)$ and $b \sim U[0, 2\pi]$.

III. GRADIENT DESCENT WITH RANDOM FOURIER FEATURES FOR SPATIO-TEMPORAL HAWKES PROCESSES

In this section, we describe our method to express (16) in terms of $\theta$ using Random Fourier features, and obtain derivative calculations for gradient descent-based optimization.

A. Random Fourier Features for Kernel Calculations

The expressions raised by Gaussian kernel inside the base and triggering intensity functions can be formed again using random Fourier features. For given two locations $s_i = (x_i, y_i)$ and $s_j = (x_j, y_j)$, the result of Gaussian kernel output in (10) can be approximated using the following $D$ dimensional random Fourier approximation [25] as

$$g_2(s_i, s_j) \approx \frac{1}{2\pi} \tilde{\Sigma}^{-1/2} z_i^T z_j,$$

where $z_i^T = \sqrt{\frac{D}{2}} \cos(s_i^T F + b^T)$ with $F \sim \mathcal{N}(0, \tilde{\Sigma})$ and $b \sim U[0, 2\pi]$. This technique embeds kernel inputs $(x, y)$ into a $D$-dimensional Euclidean inner product space using a transformation matrix $F \in \mathbb{R}^{D \times D}$ and approximates $k(x, y)$ through the inner product of embedded vectors. Although it is widely used to scale up kernel based methods such as SVM for large datasets, [29], [30] we use it to replace complex kernel calculations with straightforward methods such as SVM for large datasets.

To analyze the behavior of the approximation made by random Fourier features given in (19), we conduct a Gaussian kernel with $\sigma_x = 3$, $\sigma_y = 1$ and $\rho = 0.8$, and perform three approximations with various embedding dimensions ($D = 20, 50, 100$). We visualize the results in Fig. 2. As the number of dimensions in random Fourier features increase, approximation becomes more accurate. In the cases when $D$ is small as in Fig. 2a, some random repeating artifacts are visible around the kernel.

Since $\Sigma$ is a positive definite and symmetric matrix as mentioned in the previous section, $\tilde{\Sigma}$ is also positive-definite and symmetric. Therefore, we can decompose $\tilde{\Sigma}$ using the Cholesky decomposition, and express as

$$\tilde{\Sigma} = \tilde{C} \tilde{C}^T,$$

where $\tilde{\Sigma}$ is a unique, invertible, lower triangular $2 \times 2$ matrix with real, and positive diagonal entries.

Using the decomposition of $\tilde{\Sigma}$ in (20), we obtain the following form for vector embedding:

$$z_i^T = \sqrt{\frac{2}{D}} \cos(s_i^T \tilde{C} U + b^T),$$

where $U \sim \mathcal{N}(0, I)$. It should be emphasized that $\tilde{C}$ introduces some numerical constraints due to its properties such as being lower triangular and having positive, real diagonal entries that should be considered during optimization. To handle this issue, we use eigendecomposition of the covariance matrix $\Sigma$ to express $\tilde{C}$ in a simpler form still with constraints but more straightforward.

![Fig. 2: (a) Gaussian kernel with $\sigma_x = 3$, $\sigma_y = 1$ and $\rho = 0.8$, (b)-(c)-(d) Approximated kernels with 20, 50 and 100-dimensional random Fourier features](image-url)
ward to handle:
\[
\Sigma^{-1} = (V\Lambda V^T)^{-1},
\]
where \( \bar{C} = V\Lambda^{-1/2} \).

Now, we have two components: the eigenvector matrix \( V \in \mathbb{R}^{2 \times 2} \) with orthogonality, and the diagonal matrix of eigenvalues \( \Lambda \in \mathbb{R}^{2 \times 2} \) with positivity constraints. These components can be interpreted as the descriptors of the direction and magnitude of excitation caused by an event. As a result, we obtain the following final form for the vector embedding:
\[
z_i^T = \sqrt{\frac{2}{D}} \cos(s_i^T V\Lambda^{-1/2} U + b^T)
\]

We can express \(|\Sigma|^{-1/2} \) in (19) in terms of the diagonal elements of \( \Lambda \) as well:
\[
|\Sigma|^{-1/2} = \frac{1}{\sqrt{\prod_{i=1}^2 \ell_i}}
\]
where \( \ell = [\Lambda_{11}, \Lambda_{22}]^T \) is the vector that consists of the diagonal elements of \( \Lambda \). Since any term including covariance matrices \( \Sigma^{(\mu)} \) and \( \Sigma^{(\gamma)} \) or their corresponding Cholesky components can be expressed using \( V^{(\mu)}, V^{(\gamma)}, \ell^{(\mu)}, \) and \( \ell^{(\gamma)} \), we update the notation given for the parameter set \( \theta \) as
\[
\theta = \{K^{(\mu)}, K^{(\gamma)}, W, V^{(\mu)}, V^{(\gamma)}, \ell^{(\mu)}, \ell^{(\gamma)}\}.
\]

B. Matrix Formulations

Using the random Fourier features as described in (19), base and triggering conditional intensity function values for the \( i \) th event can be factorized as
\[
\mu_{u_i}(s_i) = \frac{1}{\bar{T}} z_i^{(\mu)^T} N^{(\mu)}(T) k^{(\mu)}_{u_i},
\]
\[
\gamma_{u_i}(t_i, s_i) = z_i^{(\gamma)^T} N^{(\gamma)}(t_i) k^{(\gamma)}_{u_i},
\]
where
\[
N^{(\mu)}(T) := \frac{1}{2\pi} |\Sigma|^{-1/2} \bar{Z}^{(\mu)}_i \ Y_{J(t)},
\]
\[
N^{(\gamma)}(t_i) := \frac{1}{2\pi} |\Sigma|^{-1/2} \bar{Z}^{(\gamma)}_i \ \text{diag}(d_{J(t_i)}) Y_{J(t)},
\]
and \( k^{(\gamma)}_{u_i} \) is the \( u_i \)th column of \( K^{(\gamma)} \), which contains the effects of other event types over the event type of \( i \) th event. In addition, we use \( J(t) := \{j|t_j < t\} \) to note the rows that belong to the events occurred before \( t \).

In order to clarify the contents of \( N^{(\mu)}(T), N^{(\gamma)}(t_i), d_{J(t_i)} \) and \( Y_{J(t_i)} \), we give their open-form expressions below:
\[
N^{(\mu)}(T) = \begin{bmatrix}
\vdots & |\Sigma^{(\mu)}|^{-1/2} \sum_{u_j=k} z_j^{(\mu)} & \ldots \\
\end{bmatrix},
\]
\[
N^{(\gamma)}(t_i) = \begin{bmatrix}
\vdots & |\Sigma^{(\gamma)}|^{-1/2} \sum_{t_j<t_i} e^{-w_{u_j u_i} (t_i-t_j) \gamma_j} & \ldots \\
\end{bmatrix},
\]
\[
Z^{(\gamma)}_{J(t_i)} = \begin{bmatrix}
\vdots & z_j^{(\gamma)^T} & \ldots \\
\end{bmatrix},
\]
\[
d_{J(t_i)} = \begin{bmatrix}
\vdots & w_{u_j u_i} \exp(-w_{u_j u_i} (t_i-t_j)) & \ldots \\
\end{bmatrix},
\]
\[
Y_{J(t_i)} = \begin{bmatrix}
\vdots & y_j^T & \ldots \\
\end{bmatrix},
\]
where \( y_j^T \) is the one-hot vector form of event type for the \( j \) th event.

Finally, using (30) and (31), the conditional intensity values for \( \mathcal{E} \) given in (16) can be expressed in the following matrix form:
\[
\begin{bmatrix}
\vdots & \Lambda(t_i, s_i) & \ldots \\
\end{bmatrix} = Q^{(\mu)} K^{(\mu)} + Q^{(\gamma)} K^{(\gamma)},
\]
where
\[
Q^{(\mu)} := \begin{bmatrix}
\vdots & -\frac{1}{\bar{T}} z_i^{(\mu)^T} N^{(\mu)}(T) & \ldots \\
\end{bmatrix}
\]
contains the relation between \( i \) th event and other events for base intensity, and
\[
Q^{(\gamma)} := \begin{bmatrix}
\vdots & -z_i^{(\gamma)^T} N^{(\gamma)}(t_i) & \ldots \\
\end{bmatrix}
\]
contains the relation between \( i \) th event and past events for triggering intensity at each row.

Once obtaining the matrix-form expression for conditional intensity in (37), we analytically derive the integral output to obtain the closed-form expression for the second term in (16)
as

\[
\Lambda_\lambda(t_i) = \frac{U}{\pi} \int_{t_{i-1}}^{t_i} \int_{s' \in S} \lambda_u(t', s') ds' dt' \\
\approx \frac{U}{\pi} \int_{t_{i-1}}^{t_i} \int_{s' \in \mathbb{R}^2} \lambda_u(t', s') ds' dt' + \sum_{u'=1}^{U} \int_{t_{i-1}}^{t_i} \int_{s' \in \mathbb{R}^2} \gamma_u(t', s') ds' dt' \\
\sim \frac{U}{\pi} \int_{t_{i-1}}^{t_i} \int_{s' \in \mathbb{R}^2} \frac{1}{T} \sum_{j=1}^{N} K_{u'j}^{(u)}(s', s_j ds' dt' \\
+ \sum_{u'=1}^{U} \int_{t_{i-1}}^{t_i} \int_{s' \in \mathbb{R}^2} \sum_{j, t_j < t'} K_{u'j}^{(u)}(t_j, t') g_2^{(u)}(s', s_j) ds' dt' \\
\approx \frac{t_i - t_{i-1}}{T} \sum_{u'=1}^{U} \sum_{j} \sum_{t_j < t'} K_{u'j}^{(u)} \\
+ \sum_{j, t_j < t} \sum_{u'=1}^{U} K_{u'j}^{(u)}(e^{-w_{u'j}t(t_i - t_j)} - e^{-w_{u'j}t(t_i - t_j)}),
\]

where we approximate \( S \) with \( \mathbb{R}^2 \) since the boundary effects will have negligible effect over the integral value. Then, the summation of \( \Lambda_\lambda(t_i) \) for consecutive events is expressed as

\[
\sum_{i=n-k}^{n} \Lambda(t_i) = \frac{t_n - t_{n-k-1}}{T} \sum_{j=1}^{U} \sum_{u'=1}^{U} K_{u'j}^{(u)} \\
- \sum_{j, t_j < t} \sum_{u'=1}^{U} K_{u'j}^{(u)}(e^{-w_{u'j}t(t_i - t_j)} \\
+ \sum_{j, t_j < t} \sum_{u'=1}^{U} K_{u'j}^{(u)}(e^{-w_{u'j}t(t_i - t_j)} \\
+ \sum_{j, t_j < t} \sum_{u'=1}^{U} K_{u'j}^{(u)}
\]

for \( 0 \leq k < i \) and \( t_0 = 0 \). Here, we utilize the relation between consecutive terms, which cancels out most of the intermediate outputs. Inserting \( n = N \) and \( k = N - 1 \) into (41) yields the following:

\[
R := \sum_{i=1}^{N} \Lambda(t_i) = \sum_{j=1}^{U} \sum_{u'=1}^{U} K_{u'j}^{(u)} + K_{u'j}^{\gamma}(1 - e^{-w_{u'j}T(t_i - t_j)}),
\]

where \( R \) is defined as the second term in (16), and has a suppressing effect over excitation matrices.

Finally, using (16), (37), and (42), we can express the negative log-likelihood as

\[
\mathcal{L} = -\text{sum}((\log \mathbf{A} \odot \mathbf{Y}) + R,
\]

where \text{sum} is the operation that sums the elements in a matrix.

### C. Derivative Calculations

In order to minimize the negative log-likelihood expressed in (43), we employ gradient descent through the back propagation of derivatives

\[
\frac{\partial \mathcal{L}}{\partial \mathbf{B}} = \left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{K}^{(u)}}, \frac{\partial \mathcal{L}}{\partial \mathbf{A}}, \frac{\partial \mathcal{L}}{\partial \mathbf{V}^{(u)}}, \frac{\partial \mathcal{L}}{\partial \mathbf{V}^{\gamma}}, \frac{\partial \mathcal{L}}{\partial \mathbf{d}_{j}}, \frac{\partial \mathcal{L}}{\partial \mathbf{d}_{j}^{(u)}} \right\}.
\]

In this section, we derive the equations for these gradients.

We can obtain the derivatives for the base and triggering intensity excitation matrices introduced in (7) and (8) as

\[
\frac{\partial \mathcal{L}}{\partial \mathbf{K}^{(u)}} = -\mathbf{Q}(41)^T (\mathbf{Y} \odot \mathbf{A}) + \frac{\partial \mathbf{R}}{\partial \mathbf{K}^{(u)}}
\]

where \( \frac{\partial \mathbf{R}}{\partial \mathbf{K}^{(u)}} \) consists of the elements \( \frac{\partial \mathbf{R}}{\partial \mathbf{K}^{(u)}_{mn}} \), which can be expressed as

\[
\frac{\partial \mathbf{R}}{\partial \mathbf{K}^{(u)}_{mn}} = \sum_{j=1}^{N} \delta_{ujm} (1 - e^{-w_{ujm}(T - t_j)}
\]

for base and triggering excitations, where \( \delta_{ij} = 1 \) if \( i = j \) and zero otherwise.

We, then, express the derivative of the decay rate matrix as

\[
\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = -\text{sum}((\frac{\partial \mathbf{A}}{\partial \mathbf{w}_{mn}})^T (\mathbf{Y} \odot \mathbf{A}) + \frac{\partial \mathbf{R}}{\partial \mathbf{w}_{mn}}.
\]

Here,

\[
\frac{\partial \mathbf{R}}{\partial \mathbf{w}_{mn}} = \sum_{j=1}^{N} \delta_{ujm} (T - t_j) e^{-w_{ujm}(T - t_j)}
\]

and \( \frac{\partial \mathbf{A}}{\partial \mathbf{w}_{mn}} \) consists of the rows

\[
\frac{\partial \mathbf{A}}{\partial \mathbf{w}_{mn}} = \frac{1}{2\pi} |\mathbf{\Sigma}^{(\gamma)}|^{-1/2} z^{(\gamma)T} \mathbf{Z}^{(\gamma)T} \mathbf{d}_{j}(41)^T \mathbf{Y} J_j(t_i) K^{(\gamma)},
\]

where

\[
\frac{\partial \mathbf{d}_{j}(t_i)}{\partial \mathbf{w}_{mn}} = \begin{bmatrix} \delta_{ujm} \delta_{ujn}(1 - w_{ujm}(t_i - t_j)) e^{-w_{ujm}(t_i - t_j)} \\
\vdots \\
\end{bmatrix}
\]

For the spatial kernel parameters, we first derive the gradients of \( \mathbf{V}^{(u)} \) and \( \mathbf{V}^{(\gamma)} \) as

\[
\frac{\partial \mathcal{L}}{\partial \mathbf{V}^{(u)}} = \left\{ \frac{\partial \mathcal{L}}{\partial \mathbf{V}^{(u)}_{mn}}, \frac{\partial \mathcal{L}}{\partial \mathbf{V}^{(u)}_{mn}} \right\},
\]

where each element is derived as

\[
\frac{\partial \mathcal{L}}{\partial \mathbf{V}^{(u)}_{mn}} = -\text{sum}((\frac{\partial \mathbf{A}}{\partial \mathbf{V}^{(u)}_{mn}})^T (\mathbf{Y} \odot \mathbf{A})
\]

Note that the derivatives of the spatial kernel parameters are not included in the above equations due to space limitations.
Here, \( \frac{\partial \mathbf{A}}{\partial V_{mn}} \) consists of the rows \( \frac{\partial a^T}{\partial V_{mn}} \) such that
\[
\frac{\partial a^T}{\partial V_{mn}} = \frac{1}{2\pi} |\Sigma(\mu)|^{-1/2} \left( \frac{\partial z_i(\mu)^T}{\partial V_{mn}} Z(\mu)^T \right) + z_i(\mu)^T \frac{\partial Z(\mu)^T}{\partial V_{mn}} \mathbf{Y}_{J(i)} \mathbf{K}(\mu)
\]
and
\[
\frac{\partial \mathbf{a}^T}{\partial V_{mn}} = \frac{1}{2\pi} |\Sigma(\gamma)|^{-1/2} \left( \frac{\partial z_i(\gamma)^T}{\partial V_{mn}} Z(\gamma)^T \right) + z_i(\gamma)^T \frac{\partial Z(\gamma)^T}{\partial V_{mn}} \mathbf{Y}_{J(i)} \mathbf{K}(\gamma)
\]
where \( \frac{\partial z_i(\mu)^T}{\partial V_{mn}} \) consists of the rows \( \frac{\partial z_i(\mu)^T}{\partial V_{mn}} = -\sqrt{\frac{2}{D}} s_i \epsilon_n^{1/2} u_{n}^T \sin(s_i^T \mathbf{V}^{(\lambda)} - 1/2 \mathbf{U} + b^T), \) and \( u_{n}^T \) is the \( n \)th row of \( \mathbf{U} \).

Finally, we obtain the derivatives of \( \mathbf{L}^{(\mu)} \) and \( \mathbf{L}^{(\gamma)} \) as
\[
\frac{\partial \mathbf{L}}{\partial \mathbf{L}} = [\cdots \frac{\partial \mathbf{L}}{\partial \mathbf{A}} \cdots]^T,
\]
where each element is defined as
\[
\frac{\partial \mathbf{L}}{\partial \mathbf{L}} = -\text{sum} \left( \frac{(\mathbf{A})^T (\mathbf{Y} \otimes \mathbf{A})}{} \right).
\]

Here, \( \frac{\partial \mathbf{A}}{\partial \mu} \) consists of the rows \( \frac{\partial a^T}{\partial \mu} \) such that
\[
\frac{\partial a^T}{\partial \mu} = \frac{1}{2\pi} \left( \frac{\partial \Sigma(\mu)}{\partial \mu} \frac{1}{2} z_i(\mu)^T Z(\mu)^T \right) + z_i(\mu)^T \frac{\partial Z(\mu)^T}{\partial \mu} \mathbf{Y}_{J(i)} \mathbf{K}(\mu)
\]
and
\[
\frac{\partial a^T}{\partial \gamma} = \frac{1}{2\pi} \left( \frac{\partial \Sigma(\gamma)}{\partial \gamma} \frac{1}{2} z_i(\gamma)^T Z(\gamma)^T \right) + z_i(\gamma)^T \frac{\partial Z(\gamma)^T}{\partial \gamma} \mathbf{Y}_{J(i)} \mathbf{K}(\gamma)
\]
where \( \frac{\partial z_i(\mu)^T}{\partial \mu} \) consists of the rows \( \frac{\partial z_i(\mu)^T}{\partial \mu} = -\sqrt{\frac{2}{D}} s_i \epsilon_n^{1/2} u_{n}^T \sin(s_i^T \mathbf{V}^{(\lambda)} - 1/2 \mathbf{U} + b^T), \) and \( u_{n}^T \) is the \( n \)th column of \( \mathbf{V}^{(\lambda)} \).

\section{D. Optimization Algorithm}

To minimize the negative log-likelihood \( \mathcal{L} \) expressed in (43), we adapt mini-batch gradient descent into our problem with a slightly modified batch generation procedure as explained in Algorithm 1. We also follow an early stopping criteria based validation method that stops the procedure if the model does not improve during \( k \) consecutive steps in terms of validation negative log-likelihood.

As mentioned before, some parameters in \( \theta \) have constraints due to the definitions of these parameters. The elements of the excitation matrices \( \mathbf{K}^{(\mu)} \) and \( \mathbf{K}^{(\gamma)} \), the decay matrix \( \mathbf{W} \), and the eigenvalue vectors of covariance matrices, \( \ell^{(\mu)} \) and \( \ell^{(\gamma)} \) have to be positive. To satisfy these conditions, we simply introduce the following intermediate variables and perform gradient descent over the unconstrained parameters \( \tilde{\mathbf{K}}^{(\lambda)} \), \( \tilde{\mathbf{W}} \) and \( \tilde{\ell}^{(\lambda)} \):
\[
\mathbf{K}^{(\lambda)} = \phi(\tilde{\mathbf{K}}^{(\lambda)}) = \frac{1}{s} \ln (1 + e^{\tilde{\mathbf{K}}^{(\lambda)})},
\]
\[
\mathbf{W} = \phi(\tilde{\mathbf{W}}) = \frac{1}{s} \ln (1 + e^{\tilde{\mathbf{W}}}),
\]
\[
\ell^{(\lambda)} = \phi(\tilde{\ell}^{(\lambda)}) = \frac{1}{s} \ln (1 + e^{\tilde{\ell}^{(\lambda)}}),
\]
where \( \phi \) is the soft-plus function parametrized by \( s \). Soft-plus function provides a differentiable and smooth approximation of ReLU function such that as \( s \rightarrow \infty \), \( \phi \rightarrow \text{ReLU} \).

Other constrained parameters are the eigenvector matrices \( \mathbf{V}^{(\mu)} \) and \( \mathbf{V}^{(\gamma)} \), which have to be orthogonal due to the eigendecomposition in (24). We employ projected gradient descent to meet this limitation by using the following update rule:
\[
\mathbf{V}^{(\mu)}_{i+1} = \Pi_\chi \left( \mathbf{V}^{(\mu)}_i - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{V}^{(\gamma)}} \right), \forall t \geq 1,
\]
where \( \chi = \{ \mathbf{X} | \mathbf{X} \in \mathbb{R}^2, \mathbf{X}^T \mathbf{X} = \mathbf{I} \} \) is the convex set of orthogonal matrices. Here, \( \Pi_\chi \) projects the updated parameter to \( \chi \) through solving the following minimization problem:
\[
\Pi_\chi (\hat{\mathbf{X}}) = \arg \min \{ \| \mathbf{X} - \hat{\mathbf{X}} \|_F \} \text{ subject to } \mathbf{X} \in \chi
\]
where \( \| \cdot \|_F \) denotes the Frobenius norm. This problem is known as Orthogonal Procrustes problem [34], and the solution yields \( \mathbf{X} = \mathbf{U} \mathbf{V}^T \), where \( \mathbf{U} \) and \( \mathbf{V} \) are obtained from \( \mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^T \) is the singular value decomposition of \( \hat{\mathbf{X}} \).

\section{IV. Experiments}

In this section, we report the results of our method in terms of fitting performance through the optimization of negative log-likelihood. In order to illustrate the results, we provide performance analysis over simulated event sequences. Then we evaluate our method in real-life datasets and compare them with other proposed methods in the literature.

\subsection{A. Synthetic Dataset Performance}

Before reporting the results of our method on real datasets, we analyze the fitting performance on synthetic data.

1) Spatio-temporal Thinning Algorithm for Simulations: In order to simulate a spatio-temporal Hawkes process, we use the thinning algorithm [35], which applies rejection sampling over pre-sampled points. We extend thinning algorithm to spatio-temporal case by sampling points distributed in time with Poisson distribution and in space with uniform distribution. Our extension closely follows from the work in [36] but
modified for spatially variant base intensity and multiple event types. The details are given in Algorithm (2).

To apply rejection sampling, we need an upper bound for the conditional intensity function, which we define as

$$\tilde{\lambda} := \max \left( \sum_{u=1}^{U} \lambda_u(t', s') \right) \text{ for } t' \in [t, +\infty) \text{ and } s' \in S$$

$$= \sum_{u=1}^{U} \left( \max(\mu_u(s')) + \sum_{j|t_j < t} K^{(\gamma)}_{u,j}(t, t_j) g^{(\gamma)}(s', s_j) \right).$$

(65)

(66)

such that $\lambda(t', s') < \tilde{\lambda}$ for all $t \geq t'$ and $s' \in S$.

Unfortunately, it is not straightforward to compute the exact value of $\tilde{\lambda}$ due to non-monotonous structure of the conditional intensity function over spatial domain. Instead, we perform calculations for densely sampled spatial points over $S$ at time $t$, and take the maximum value as our approximation. Moreover, in (66), older events will have significantly less effect over the total sum due to the exponential decay kernel. Thus, to make simulation process computationally efficient, we ignore the triggering effects of the events occurred before a particular temporal offset.

To generate the type of the thinned event in Algorithm (2), we apply the thinning procedure over the total conditional intensity and then draw the event type stochastically from the generated $p(u)$ for the generated spatio-temporal point. Instead of running rejection sampling for each event type separately, this procedure provides an efficient and convenient way to generate spatio-temporal Hawkes process with multiple event types.

We simulate realizations with $T = 100000$ and $S = [[-1, 1], [-1, 1]]$ from each of the three spatio-temporal Hawkes processes. Each process has different parameters to analyze the behavior of our model in different cases and observe its flexibility. Table I shows the parameter sets used for simulations.

<table>
<thead>
<tr>
<th>Table I: Simulation Configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>id</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td><strong>id</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

Figures 3 and 4 illustrate the output for the second simulation spatially and temporally. Simulated events clearly show a clustered pattern over spatial and temporal space. It is also possible to observe that some random events are generated from base intensity as well.

2) Fitting Performance: To analyze the fitting capability of our model, we evaluate the negative log-likelihood over the sequences of each simulation. We also evaluate two simpler versions of our model as baselines. First, we restrict conditional
and allowing the spatial kernel to rotate and have elliptic shape increases the performance. As expected, Poisson model has significantly lower performance in the simulated Hawkes process realizations due to its incapability to capture the exhibitions and inhibitions between event types.

B. Real-Life Performance

In order to illustrate the fitting performance of our method in real-world datasets, we investigate the negative log-likelihood. After learning the process parameters, we perform network analysis by examining the interactions between different event types in terms of excitation/inhibition relations and spatio-temporal effects. To this end, we have chosen the following three datasets. These datasets have been studied in the context of point processes, with applications on spatio-temporal prediction, hotspot analysis, and network reconstruction. They also exhibit some characteristics such as having clustered structure with spatiotemporally triggering relations.

1) Datasets:

a) Chicago Crime Dataset: Chicago Crime Dataset includes the reported incidents in the City of Chicago from 2001, and is still being updated by Chicago Police Department. The dataset includes the location and time of the incidents as well as their types such as theft, burglary, assault etc. Before collecting results, we grouped event types into 7 different classes (1: Theft; 2: Robbery, Burglary; 3: Narcotics; 4: Battery, Assault, Offense; 5: Arson; 6: Rape, Sex; 7: Others). Furthermore, we filter the locations spatially between the latitudes of [41.500350, 42.220382] and longitudes of [-88.095633, -87.447644] to remove outlier regions.

b) Significant Earthquakes Dataset: The National Earthquake Information Center provides this dataset that includes earthquakes with the magnitude of 5.5 or higher since 1965. Every earthquake entry includes a record of the date, time, location, depth, magnitude, and source. We filter the dataset temporally and work on the events after the year of 2000. In addition, we cluster earthquakes into three classes depending on their magnitudes.

c) Gowalla Friendship Network Dataset: Gowalla is a location-based social networking website, where users share their locations by checking-in [37]. This dataset includes a collection of 6,442,890 check-ins of these users throughout Feb. 2009 - Oct. 2010. For the experiments, we work on the data between June 2010 - Sep. 2010, and cluster users into 5 groups depending on their check-in frequency.

2) Fitting Performance: First, we investigate the fitting performance of the proposed optimization algorithm with respect to expectation-maximization (EM) algorithm and stochastic clustering, which both are widely studied in the literature of spatio-temporal Hawkes processes. To prevent numerical instability issues related to very low/high temporal/spatial space size, and report comparable results among all datasets, we scale the given event sequence spatio-temporally. In the time domain, we stretch or shrink the event times such that the average temporal distance between consecutive events becomes 1 millisecond. We perform min-max scaling in the spatial domain to map original space into location values in
between −1 and 1. In addition, we repeat the experiments 10 times to reduce the random effects on performance as much as possible. (More details will be included once they are finalized)

TABLE III: Fitting Performance in Real-world Datasets in terms of Negative Log-Likelihood

<table>
<thead>
<tr>
<th>Method</th>
<th>Chicago</th>
<th>Earthquake</th>
<th>Gowalla</th>
</tr>
</thead>
<tbody>
<tr>
<td>EM [10]</td>
<td>−0.87</td>
<td>−2.14</td>
<td></td>
</tr>
<tr>
<td>Stochastic Declustering [17]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RFF-GD (Proposed)</td>
<td>−3.18</td>
<td>−1.14</td>
<td>−2.60</td>
</tr>
</tbody>
</table>

Fig. 5: Scatterplots of Negative Log-Likelihood Comparisons

Fig. 6: Learned Excitation Matrices for Chicago (a), Earthquake (b) and Gowalla (c) Datasets

3) Network Analysis:

V. CONCLUSION

We studied spatio-temporal network analysis using Hawkes processes. We introduced novel inference method for spatio-temporal Hawkes process to avoid the limitations of conventional algorithms in the literature such as Expectation-Maximization [10] and stochastic declustering [11]. To this end, we propose a gradient descent based optimization algorithm, which utilizes random Fourier features for kernel approximations. We use reparameterization techniques and projected gradient descent to meet structural constraints of the process parameters. We analyze the improvements achieved by the proposed inference method on various simulations and three real-world datasets. The comparisons show that the proposed method performs better in terms of negative log-likelihood compared to other algorithms. Besides, we interpret the learned process parameters and perform network analysis over these real-world datasets.

REFERENCES


