Hawkes Process Multi-armed Bandits for Search and Rescue

Wen-Hao Chiang
Department of Computer & Information Science
Indiana University–Purdue University Indianapolis
Indianapolis, USA
chiangwe@iupui.edu

George Mohler
Department of Computer Science
Boston College
Chestnut Hill, USA
mohlerg@bc.edu

***Abstract*—We propose a novel framework for integrating Hawkes processes with multi-armed bandit algorithms to solve spatio-temporal event forecasting and detection problems when data may be undersampled or spatially biased. In particular, we introduce an upper confidence bound algorithm using Bayesian spatial Hawkes process estimation for balancing the trade-off between exploiting geographic regions where data has been collected and exploring geographic regions where data is unobserved. We first validate our model using simulated data. We then apply it to the problem of disaster search and rescue using calls for service data from hurricane Harvey in 2017 and the problem of detection and clearance of improvised explosive devices (IEDs) using IED attack records in Iraq. Our model outperforms state-of-the-art baseline spatial MAB algorithms in terms of cumulative reward and several other ranking evaluation metrics.**

***Index Terms*—Multi-armed bandit, Upper confidence bound, Hawkes processes, Bayesian inference, Disaster rescue**

I. Introduction

There are a variety of scenarios where a sequential set of decisions is made, each followed by some gain in information, that allows us to refine our future decisions or "strategies". Often this information may come in the form of a reward or payoff (that may be negative). Examples of such scenarios include online advertising [1] where spending can occur in a known, profitable channel or in a new, possibly better channel, personalized recommendations [12], [26] of a past product purchase or a new, possibly better product, and clinical trials [7] between an established drug and a new treatment. In each of these cases a balance must be struck between maximizing payoffs using known information on treatment units and retrieving more information from those under-sampled treatment units.

One application area where such a decision process occurs is that of disaster search and rescue. During hurricane Harvey in 2017, Houston experienced significant flooding and a number of citizens required rescue by boat. Information on when and where these rescues needed to occur resided in disparate data feeds, for example some citizens were rescued by government first responders via 911 or 311 calls, others were rescued through social media posts by the "Cajun Navy," a volunteer rescue group [27]. During disasters, a particular dataset may be over sampled in one area and undersampled in another due to power outages, cell tower outages, demographic disparities on the use of social media, etc. For a group like the Cajun Navy who relied on under-sampled social media data, along with random search, machine learning based optimal search strategies that can adapt to spatio-temporal clustering in disaster event data would be beneficial. Similar strategies are also needed in other space-time searching scenarios, such as searching for and clearing improvised explosive devices (IED). Certain regions of a road network may be over-sampled due to previously identified or detonated devices, whereas other areas may be unknown and contain IEDs yet to be detonated.

We believe multi-armed bandits (MAB) are well suited for this task of balancing geographical exploration during disaster search and rescue vs. exploiting known, biased data on locations needing help. In the classic MAB problem setup, a gambler chooses a lever to play at each round over a planning horizon and only the reward from the pulled lever is observed. The gambler's goal is to maximize the total reward while using some trials (with negative payoff) to improve understanding of the distribution of the under-observed levers. For the disaster search and rescue scenario, each geographic region and window of time may be viewed as a lever, where information may be known about areas previously visited or having historical data, but is not known about other areas. Here the reward is discovery of a citizen needing rescue. We note there has been past research on mining spatio-temporal disaster data. Some research has been dedicated to disaster mitigation and management in order to minimize casualties [11]. Data mining tasks include but are not limited to decision tree modeling for flood damage assessment [19], statistical model ensembles for susceptible flood regions prediction [28], and text mining social media for key rescue resource identification [4]. However, no work to date has tackled the problem from a MAB framework. Furthermore, there are few existing spatial MAB algorithms and, to our knowledge, no MAB algorithm has been developed for data exhibiting clustering both in space and time.

For this purpose we introduce the Hawkes process multi-armed bandit. The method is capable of detecting spatio-temporal clustering patterns in data, while capturing uncertainty of estimated risk in under-sampled geographical regions. The output of the model is a decision strategy for optimizing spatio-temporal search and rescue decisions. The outline of the

paper is as follows. In Section II we review relevant literature on multi-armed bandits. In Section III we present our spatio-temporal MAB problem formulation and introduce the Hawkes process multi-armed bandit methodology. In Section IV we conduct several experiments on both synthetic and real data illustrating the advantages of our approach over existing MAB baseline models.

II. Background on multi-armed bandits

Here we review existing literature on multi-armed bandits (MAB). Several categories of algorithms exist including ϵ-greedy, Bayes rule, and upper confidence bound algorithms. In the case of ϵ-greedy algorithms, many adaptations have been proposed. The authors in [29] follow a probability distribution to select levers during the exploration phase, and such probability is calculated through a softmax function, where a "temperature" parameter is introduced to adjust how often random actions are chosen during exploration. In the line of work from [30], budgets for pulling levers are further considered. Levers are uniformly pulled within the budget limit during the first ϵ rounds (i.e., exploration phase). In the rest of $1-\epsilon$ rounds, they then solve the exploitation optimization as an unbounded knapsack problem by viewing costs, values, and budgets as weights, estimated rewards, and knapsack capacity, respectively [30]. One of the most popular approaches in the Bayes rule family of MAB problems is Thompson sampling [10]. It starts with a fictitious prior distribution on rewards, and the posterior gets updated as actions are played. While in some cases sampling from the complex posterior may be intractable, [8] replace the posterior distribution by a bootstrap distribution. In the work of [13], they focus on scenarios with drifting rewards and tackle such a problem by assigning larger weights to more recent rewards when updating the posterior distribution.

Finally, upper confidence bound (UCB) algorithms [17] are one of the most popular strategies in MAB. Essentially, a UCB algorithm builds a bounded interval that captures the true reward with high possibility, and levers with higher bounds tend to be selected. UCB algorithms are also widely studied within the setting of contextual bandit problems where rewards or actions are characterized by features (i.e., context). Given the observations on the features of rewards, the authors in [18] and [6] model the reward function through linear regression, and the predictive reward is further bounded by predictive variance. Such an idea is shared by [16] where the authors adopt Gaussian process regression ($\mathcal{GP}$) to bound the predicted rewards by the posterior mean and standard deviation conditioned on the past observations. In the recent work of [31], they take a further step by encoding geolocation relations between levers into features when rewards are collected in a domain of space. Even though $\mathcal{GP}$ regression modeling takes spatial relations among levers into account [31], the lack of consideration for non-stationary rewards or temporal clustering patterns make it inapplicable in many real-world problems such as those considered in this paper. To overcome such shortcomings, we develop an upper confidence bound strategy using Bayesian Hawkes processes ($\mathcal{HP}$s) in this work. $\mathcal{HP}$s have been widely studied and applied in many areas from earthquake modeling [9] and financial contagion [3], to event spike prediction [5] and crime prevention [20]. However, $\mathcal{HP}$s have not been combined with MAB strategies to date and we show how they can be seamlessly integrated with existing UCB algorithms to build a spatial and temporal aware MAB algorithm.

III. Methodology

Here we provide the details of our MAB Hawkes process methodology. We view each sub-region of space as a MAB "lever" and the count of disaster events observed in a pulled lever as the reward. In each round we select several sub-regions to search and our goal is to maximize the cumulative number of events observed over time in pulled levers.

A. Spatio-temporal MAB problem formulation

We first partition the entire spatial domain into a set of grid cells, and we denote this set of cells as $\mathcal{A} = \{a_1, a_2, \cdots\}$. We divide the range of longitude and latitude evenly into X and Y grids, i.e., $X \times Y$ cells in total. Each grid cell is characterized by a feature vector $\mathbf{x_a}$. In this manuscript, we use the grid indicators as features to describe the geolocation of a cell, i.e., $\mathbf{x_a} = [\mathrm{x}, \mathrm{y}]^\intercal$. Given a time span T, each multi-armed bandit (MAB) algorithm recommends a short ranked list consisting of N cells to visit, denoted as $\boldsymbol{a}$, for every W time units. For each visit v at cell $a \in \boldsymbol{a}$, we observe the events that occurred in the cell, denoted as $\mathcal{T}_v^a$, and we consider the number of discovered events $|\mathcal{T}_v^a|$ as rewards r_v^a. Our goal is to maximize total rewards, i.e., the total number of observed events in the visited cells after a total of V visits. This type of sequential decision-making task is a spatio-temporal multi-armed bandit problem in which each cell is viewed as a lever, and each visit to the set of chosen grid cells (constrained by resources) can be viewed as pulling the levers of the MAB machine.

B. Hawkes Process multi-armed bandits

We model the occurrence of events in space and time using a Hawkes process where the intensity is given by,

$$\lambda(t|\theta, \mathcal{T}) = \mu + \sum_{t_i < t, t_i \in \mathcal{T}} \alpha\beta \exp^{-\beta(t-t_i)}. \tag{1}$$

Here, θ represents the parameters (μ, α, β) where μ is the background intensity; α is the infectivity factor (when viewed as a branching process this is the expected number of direct offspring an event triggers); and β is the exponential decay rate capturing the time scale between generations of events. Here $\mathcal{T}$ is the set of timestamps for inference.

At each round of the multi-armed bandit (MAB) process, we select N cells with the highest estimated risk to visit, and we observe the events. However, time have elapsed between consecutive visits to a cell and there is a gap that needs to be filled. Therefore, to fill up these gaps, we simulate Hawkes processes ($\mathcal{HP}$s) by thinning [2] based on the inferred parameters. A combination of actual observations and simulated

events is then defined as a set of timestamps that represents our best guess on the missing gap for each grid cell. We denote these sets of timestamps as $\hat{\boldsymbol{\mathcal{S}}} = \{\hat{\mathcal{S}}^a | a \in \mathcal{A}\}$. After each visit, we update each $\hat{\mathcal{S}}^a$ by choosing the most likely $\mathcal{HP}$ realization and defining that as the event history.

To estimate the Hawkes process parameters, we use Bayesian inference * to estimate $\hat{\mathcal{S}}^a$ for each visited cell $a \in \boldsymbol{a}$ and to estimate the parameters of $\mathcal{HP}$s [24]. The likelihood function is given by Equation 2 , where $\hat{\mathcal{S}}^a = \{t_1, t_2, \cdots, t_n\}$.

$$\mathcal{L}(\hat{\mathcal{S}}^a | \mu, \alpha, \beta) = \prod_{i=1}^{|\hat{\mathcal{S}}^a|} \lambda(t_i) \text{exp}^{-\int_0^{t_n} \lambda(u) du}. \quad (2)$$

If we denote the prior by $p(\mu, \alpha, \beta)$, we get the posterior $p(\mu, \alpha, \beta | \hat{\mathcal{S}}^a) \propto p(\mu, \alpha, \beta) \mathcal{L}(\hat{\mathcal{S}}^a | \mu, \alpha, \beta)$, where $0 < \mu, \beta < \infty$ and $0 < \alpha < 1$. Here, we choose a gamma distribution ($\mathcal{G}$) as a prior for μ and β, and we choose a beta distribution ($\mathcal{B}$) as a prior for α. That is, $p(\mu),\ p(\beta) \sim \mathcal{G}(k_p, k_c)$, where k_p and k_c are the shape and scale parameter for $\mathcal{G}$, respectively ; and $p(\alpha) \sim \mathcal{B}(m, n)$, where m and n are both shape parameters for $\mathcal{B}$.

We use Metropolis-Hastings [25] to draw samples from the posterior distribution. We then denote such a set of parameters as $\boldsymbol{\Theta} = \{\theta_1, \theta_2, \cdots, \theta_L\}$, where $\theta_l = (\mu_l, \alpha_l, \beta_l)$. For each θ_l, we simulate a $\mathcal{HP}$ realization and denote them as $\widetilde{\mathcal{S}}^a = \{\widetilde{\mathcal{S}}^a_l | l = 1, 2, \cdots, L\}$. Together with all $\widetilde{\mathcal{S}}^a$ where $a \in \mathcal{A}$, we denote them as $\widetilde{\boldsymbol{\mathcal{S}}}$. Note that the base intensity is a function of time, i.e., $\mu(t)$, contributed by the best guess $\hat{\mathcal{S}}_a$. Given the newly observed timestamps $\mathcal{T}^a_v$, together denoted as $\boldsymbol{\mathcal{T}}_v = \{\mathcal{T}^a_v | a \in \boldsymbol{a}\}$, we then fill up the gap between the best guess $\hat{\mathcal{S}}^a$ and the observed timestamps by selecting the set of simulated timestamps, denoted as $\widetilde{\mathcal{S}}^a_{\hat{l}}$, where $\mathcal{T}^a_v$ has the largest likelihood as in Equation 3. Finally, we update our best guess for observed cells (i.e., $\hat{\mathcal{S}}^a$) as in Equation 3.

$$\hat{l} = \underset{l}{\text{argmax}}\, \mathcal{L}(\mathcal{T}^a_v | \theta_l, \{\widetilde{\mathcal{S}}^a_l, \hat{\mathcal{S}}^a\}); \quad \hat{\mathcal{S}}^a = \{\mathcal{T}^a_v, \widetilde{\mathcal{S}}^a_{\hat{l}}, \hat{\mathcal{S}}^a\} \quad (3)$$

C. Spatial Upper Confidence Bound on Event Intensities

In this section we show how to incorporate the spatial relationships between cells and build an upper confidence bound (UCB) on event intensities. For each cell, we have a set of simulated timestamps $\widetilde{\boldsymbol{\mathcal{S}}}$, and we can estimate the event intensities up to current time t_c through the intensity function in Equation 1. The set of event intensities of each cell a are denoted as $\boldsymbol{\lambda}^a(t_c) = \{\lambda^a_l(t_c | \theta_l, \mathcal{T}^a_l) |\ l = 1, 2, \cdots, L\}$, where $\mathcal{T}^a_l = \{\widetilde{\mathcal{S}}^a_l, \hat{\mathcal{S}}^a\}$ and $\theta_l = (\mu_l, \alpha_l, \beta_l)$.

Inspired by the UCB algorithm, we consider its ζ_{hp} standard deviation above the mean as the UCB on the intensities:

$$s^a_{\text{hp}} = \overline{\boldsymbol{\lambda}}^a(t_c) + \zeta_{\text{hp}} \times \sigma_{\boldsymbol{\lambda}^a(t_c)}, \quad (4)$$

where $\overline{\boldsymbol{\lambda}}^a(t_c)$ and $\sigma_{\boldsymbol{\lambda}^a(t_c)}$ are the mean and the standard deviation of estimated intensities in $\widetilde{\mathcal{S}}^a \in \widetilde{\boldsymbol{\mathcal{S}}}$; ζ_{hp} is the parameter to decide how much we look at the upper bound when selecting the cells based on the estimated intensities; and s^a_{hp} is the UCB on the intensities. Together, we define it as $\boldsymbol{s}_{\text{hp}} = \{s_{\text{hp}} | a \in \mathcal{A}\}$.

For $\boldsymbol{\lambda}^a(t_c)$, the last visit in each cell is varied, and thus, the time span for each simulation $\widetilde{\mathcal{S}}^a_l$ is different as well. To smooth out the impact from the variations, we apply a 2D Gaussian smoothing filter ($\mathcal{GF}$) across all grid cells and incorporate the spatial relationships between cells by taking $\mathbf{x_a} = [x, y]^\intercal$ into account for the coefficient calculation. In particular, a 2D $\mathcal{GF}$ modifies the $\boldsymbol{s}_{\text{hp}}$ by the convolution with a Gaussian function $g(x, y) = \frac{1}{2\pi\sigma^2_{\text{hp}}} \text{exp}(-\frac{x^2+y^2}{2\sigma^2_{\text{hp}}})$, where σ_{hp} is the standard deviation of the Gaussian distribution. Such a smoothing process is defined as $\mathcal{GF}(\boldsymbol{s}_{\text{hp}} | \sigma_{\text{hp}})$, and the smoothed UCB on intensities is then defined as $\overline{\boldsymbol{s}}_{\text{hp}} = \{\overline{s}^a_{\text{hp}} | a \in \mathcal{A}\}$. In Figure 1, we present the framework for determining a cell score $\overline{s}^a_{\text{hp}}$.

D. Baseline Methods

We will compare the Hawkes process MAB to several baseline algorithms. We also show in the subsequent section how to combine existing MAB strategies with the Hawkes process MAB to further improve performance.

1) Epsilon Greedy ϵ`Gdy`*::* The epsilon-greedy algorithm [17] separates trials into exploitation and exploration phases using a proportion of ϵ and $1 - \epsilon$ visits respectively. We keep track of the average reward per visit for every cell after each visit, and then we visit the cells with the highest average reward during exploitation. During the exploration phase, we visit all the cells uniformly at random.

2) Upper Confidence Bound `UCB1`*::* For upper confidence bound (UCB) algorithm [17], we construct an UCB on rewards so that the true value is always below the UCB with a high probability. We then pay visits to those promising cells with the highest UCB. The UCB of cell a during visit v is defined as a score, s^a_{ucb}, calculated as $s^a_{\text{ucb}} = \overline{r^a_v} + \zeta_{\text{ucb}} \sqrt{\frac{2 \log v}{n^a_v}}$, where $\overline{r^a_v}$ is the average reward per visit and n^a_v is the total number of visits up to visit v. The parameter ζ_{ucb} is to control how optimistic we are during the processes. Finally, after each visit, we update the scores s^a_{ucb}, and we select the top-N cells with the highest score for the next visit.

3) Spatial Upper Confidence Bound `SpUCB`*::* While the epsilon-greedy algorithm considers the cells with the largest mean value of rewards during exploitation, and the upper confidence bound (UCB) algorithm selects the most optimistic cells, neither considers the spatial relationship between the cells and the corresponding events. To introduce such geolocation information into MABs, [31] propose a space-aware UCB algorithm utilizing a Gaussian Process regression model ($\mathcal{GP}$) [23] and building up a spatial UCB from the predicted expectation and uncertainty for each lever. After each visit, we collect the features of visited cells and their corresponding rewards, denoted as $\mathcal{X}$ and $\mathcal{Y}$, respectively. Together with previous collections, i.e., $\mathcal{X} = \mathcal{X} \cup \{\boldsymbol{x}_a | a \in \boldsymbol{a}\}$ and $\mathcal{Y} = \mathcal{Y} \cup \{r_a | a \in \boldsymbol{a}\}$, we train the $\mathcal{GP}$ regression model. In the $\mathcal{GP}$, we hold a prior assumption that the correlations between two cells a and a' slowly decay following an exponential

*https://github.com/canerturkmen/hawkeslib

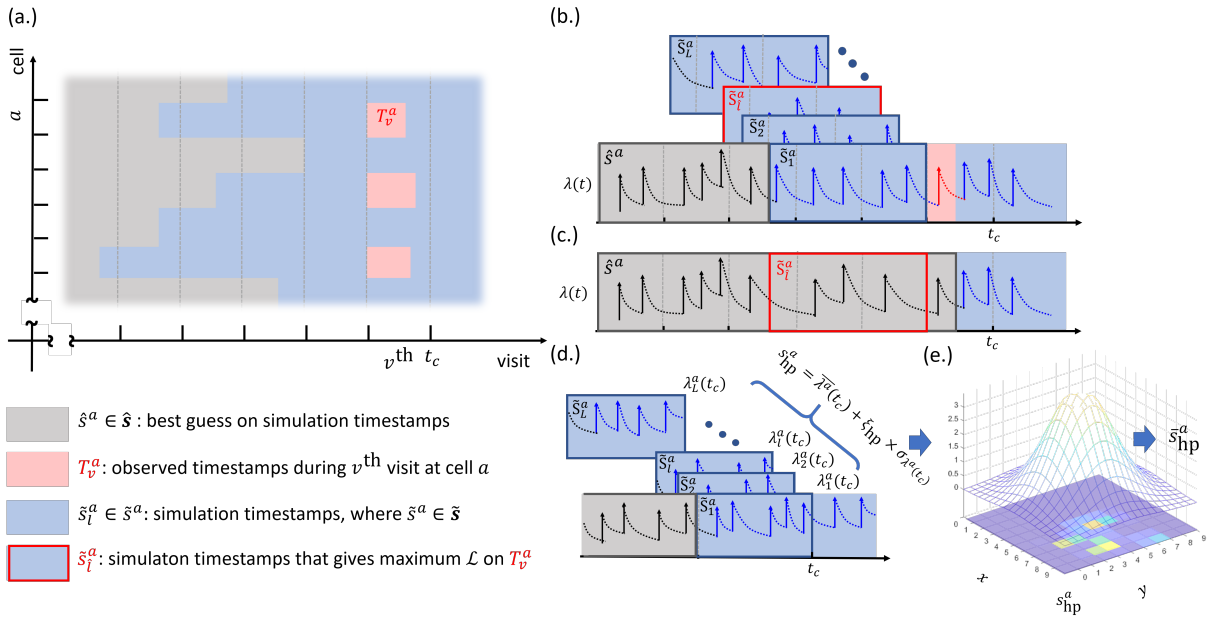

Fig. 1: Overall framework on score $\overline{\boldsymbol{s}}_{\text{hp}}$ in HpSpUCB.

function of their distance. Thus, we select a radial basis kernel, $\mathbf{k}_{\text{RBF}}$, as the covariance of a prior distribution over the target functions. The kernel function $\mathbf{k}_{\text{RBF}}$ is calculated as follows: $\mathbf{k}_{\text{RBF}}(\boldsymbol{x}_a, \boldsymbol{x}_{a'}) = \exp\left(\frac{-\|\boldsymbol{x}_a, \boldsymbol{x}_{a'}\|^2}{2\sigma_{\text{gp}}^2}\right)$, where σ_{gp} is a parameter that determines how far the correlation extends.

After each visit, we build the spatial UCB based on the prediction for each cell a by looking at its ζ_{gp} predicted uncertainty above the expected mean, and we denote such a UCB as s_{gp}^a (Equation 5). Here, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are the predicted expectations and uncertainties given cell a and ζ_{gp} governs how far we expend our upper confidence bound. Unlike ϵ-Gdy and UCB1 in which only cells with the largest score are selected, the recommended cells are sampled for the next visit without replacement based on a probability distribution. Such probability distribution p_{gp}^a is calculated by a softmax function on s_{gp}^a as in Equation 5, where τ_{gp} can be viewed as a temperature parameter that adjusts the exploitation and exploration ratio. We further denote such a baseline method as SpUCB.

$$s_{\text{gp}}^a = \boldsymbol{\mu}(\boldsymbol{x}_a) + \zeta_{\text{gp}}\boldsymbol{\sigma}(\boldsymbol{x}_a); \quad p_{\text{gp}}^a = \frac{\exp(s_{\text{gp}}^a/\tau_{\text{gp}})}{\sum_{a\in\mathcal{A}}\exp(s_{\text{gp}}^a/\tau_{\text{gp}})}. \quad (5)$$

E. Combining Hawkes Process Bandits with Existing Methods

Even though the upper confidence bound (UCB) built upon the Hawkes process ($\mathcal{HP}$) can track the event intensities, here we show how to improve its accuracy during the early stages of the multi-armed bandit (MAB) process. We combine the score from the UCB on intensities, $\overline{s}_{hp}^a$ in $\overline{\boldsymbol{s}}_{\text{hp}}$, with the score from the previously introduced method, that is, s_{ucb}^a from UCB1 or s_{gp}^a from SpUCB, respectively. We denote the combined score as $\widehat{s}^a$. Finally, we use a softmax function to calculate the probability $\widehat{p}^a$, and sample N cells without replacement based on the probability for our next visit. We then denote our model as HpSpUCB.

More specifically, $\widehat{s}^a$ and $\widehat{p}^a$ are calculated as in Equation 6 where γ governs how much we rely on intensities estimated through $\mathcal{HP}$s, and we can adjust our model based on how much the dataset itself contains a self-excitation pattern. Note that we define $\widehat{s}^a$ and $\widehat{p}^a$ from all cells as $\widehat{\boldsymbol{s}}$ and $\widehat{\boldsymbol{p}}$, respectively.

$$\widehat{s}^a = s^a + \gamma\overline{s}_{\text{hp}}^a; \quad \widehat{p}^a = \frac{\exp(\widehat{s}^a/\tau)}{\sum_{a\in\mathcal{A}}\exp(\widehat{s}^a/\tau)}. \quad (6)$$

Based on the different choices of s^a to combine with the $\mathcal{HP}$ component, we have two variations as our proposed models for comparison:

1) $\text{UCB1}_{\text{HpSp}}$ where $s^a = s_{\text{ucb}}^a$, that is, we combine $\overline{s}_{\text{hp}}^a$ with scores from UCB1;
2) HpSpUCB where $s^a = s_{\text{gp}}^a$, that is, we combine $\overline{s}_{\text{hp}}^a$ with scores from SpUCB.

Note that with HpSpUCB and $\text{UCB1}_{\text{HpSp}}$, we can compare when we choose different models to incorporate the proposed $\mathcal{HP}$ component. The overall MAB process of HpSpUCB is presented in the algorithm 1, and algorithm 2 shows how our $\mathcal{HP}$ component plays its part in HpSpUCB.

IV. Experiments

A. Datasets

1) Spatial-temporal Synthetic Data ($\mathcal{D}_{Syn}$):: We first validate our methodology using a simulated Hawkes process ($\mathcal{HP}$) [21]. We generate a synthetic dataset by first simulating a Poisson process for initial immigrant events, of which the average number follows a Poisson distribution $\mathcal{P}(\eta T)$, and distribute them uniformly in space. Note that η is the rate per second and T is the total time span. Next, we generate a Poisson process recursively for each event in each generation by the following steps:

Algorithm 1 Algorithm of HpSpUCB

procedure HPSPUCB($\mathcal{A}$, γ, ζ_{hp}, ζ_{gp}, σ_{gp}, τ)
  $X \leftarrow \emptyset$, $\boldsymbol{y} \leftarrow \emptyset$, $\hat{\boldsymbol{\mathcal{S}}} \leftarrow \emptyset$, $\widetilde{\boldsymbol{\mathcal{S}}} \leftarrow \emptyset$, $t_c \leftarrow W$, $\boldsymbol{a} \leftarrow$ select N cells at random
  for $v = 1$ to V **do** ▷ MAB process
    $\boldsymbol{\mathcal{T}} \leftarrow \{\mathcal{T}_v^a | a \in \boldsymbol{a}\}$,
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{\boldsymbol{x}_a | a \in \boldsymbol{a}\}$, $\mathcal{Y} \leftarrow \mathcal{Y} \cup \{r_a | a \in \boldsymbol{a}\}$
    $\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow \mathcal{GP}(\mathcal{X}, \mathcal{Y} | \sigma_{\text{gp}})$ ▷ model and infer for $\mathcal{GP}$
    $\boldsymbol{s}_{\text{gp}} \leftarrow \{s_{\text{gp}}^a = \boldsymbol{\mu}(\boldsymbol{x}_a) + \zeta_{\text{gp}} \boldsymbol{\sigma}(\boldsymbol{x}_a) | a \in \mathcal{A}\}$
    $\overline{\boldsymbol{s}}_{\text{hp}}, \hat{\boldsymbol{\mathcal{S}}}, \widetilde{\boldsymbol{\mathcal{S}}} \leftarrow$ HPUCB($\hat{\boldsymbol{\mathcal{S}}}$, $\widetilde{\boldsymbol{\mathcal{S}}}$, $\boldsymbol{a}$, σ_{gp}, $\zeta_{\text{hp}}\, t_c$, $\mathcal{A}$, $\boldsymbol{\mathcal{T}}$)
    $\widehat{\boldsymbol{s}} \leftarrow \{\widehat{s}^a = s_{\text{gp}}^a + \gamma \overline{s}_{\text{hp}}^a | a \in \mathcal{A}\}$
    $\widehat{\boldsymbol{p}} \leftarrow$ apply softmax function on $\widehat{\boldsymbol{s}}$
    $\boldsymbol{a} \leftarrow$ sample N cells based on probability $\widehat{\boldsymbol{p}}$
    $t_c \leftarrow (v+1) \times W$ ▷ update the current time t_c
  end for
end procedure

Algorithm 2 Calculation of $\overline{s}_{\text{hp}}^a$

function HPUCB($\hat{\boldsymbol{\mathcal{S}}}$, $\widetilde{\boldsymbol{\mathcal{S}}}$, $\boldsymbol{a}$, σ_{gp}, $\zeta_{\text{hp}}\, t_c$, $\mathcal{A}$, $\boldsymbol{\mathcal{T}}$)
  for all $a \in \boldsymbol{a}, \hat{\mathcal{S}}^a \in \hat{\boldsymbol{\mathcal{S}}}, \widetilde{\mathcal{S}}^a \in \widetilde{\boldsymbol{\mathcal{S}}}$ **do**
    $\hat{l} \leftarrow \text{argmax}_l \mathcal{L}(\mathcal{T}_v^a | \theta_l, \{\hat{\mathcal{S}}^a, \widetilde{\mathcal{S}}_l^a\})$, where $\widetilde{\mathcal{S}}_l^a \in \widetilde{\mathcal{S}}^a$
    $\hat{\mathcal{S}}^a \leftarrow \{\hat{\mathcal{S}}^a, \mathcal{S}_{\hat{l}}^a, \mathcal{T}_v^a\}$
    $\Theta \leftarrow p(\mu, \alpha, \beta | \hat{\mathcal{S}}_a)$
    $\widetilde{\mathcal{S}}^a \leftarrow \{\widetilde{\mathcal{S}}_l^a = \mathcal{HP}(\hat{\mathcal{S}}^a | \theta_l) | \theta_l \in \Theta\}$
    update $\hat{\mathcal{S}}^a$ in $\hat{\boldsymbol{\mathcal{S}}}$ and $\widetilde{\mathcal{S}}^a$ in $\widetilde{\boldsymbol{\mathcal{S}}}$
  end for
  $\boldsymbol{s}_{\text{hp}} \leftarrow \{s_{\text{hp}}^a = \overline{\boldsymbol{\lambda}}^a(t_c) + \zeta_{\text{hp}} \times \sigma_{\boldsymbol{\lambda}^a(t_c)} |\ a \in \mathcal{A}\}$, where $\boldsymbol{\lambda}^a(t_c) = \{\lambda_l^a(t_c | \theta_l, \mathcal{T}_l^a) |\ l = 1, 2, \cdots, L\}$ and $\mathcal{T}_l^a = \{\widetilde{\mathcal{S}}_l^a \cup \hat{\mathcal{S}}^a \ | \widetilde{\mathcal{S}}_l^a \in \widetilde{\mathcal{S}}^a, \hat{\mathcal{S}}^a \in \hat{\boldsymbol{\mathcal{S}}}\}$
  $\overline{\boldsymbol{s}_{\text{hp}}} \leftarrow \mathcal{GF}(\boldsymbol{s}_{\text{hp}} | \sigma_{\text{hp}})$
  return $\overline{\boldsymbol{s}_{\text{hp}}}, \hat{\boldsymbol{\mathcal{S}}}, \widetilde{\boldsymbol{\mathcal{S}}}$
end function

1) Draw a sample following a Poisson distribution $\mathcal{P}(\phi)$ as the number of offspring, where ϕ governs the average number of offspring that an event spawns;
2) Sample the waiting time between parent and offspring following an exponential distribution $\mathcal{E}(\omega)$;
3) Sample the spatial distance between parent and offspring event according to a normal distribution $\mathcal{N}(\sigma)$; and
4) Accept and record the event only when it is within the domain of time and space. We then go back step 1. and move on to the next recent event.

The simulation stops when all of a generation are outside T. We then denote these synthetic datasets as $\mathcal{D}_{\text{Syn}}$. In Figure 2, we present a realization of the synthetic data in $\mathcal{D}_{\text{Syn}}$ and show the top-20 largest clusters generated by the immigrants.

2) City of Houston 311 Service Requests ($\mathcal{D}_{Hry}$):: We apply the methodology to geolocated Houston 311 calls for service with time and geo-location labels during the time period of hurricane Harvey in 2017 in Houston (08/23/2017 to 10/02/2017). [†]. Among all kinds of services, we focus on "flooding" events that contain complete timestamp, longitude and latitude information. In total, there are $4,315$ 311 flooding events within Houston, Texas, and we denote this dataset as

[†]http://hfdapp.houstontx.gov/311/311-Public-Data-Extract-Harvey-clean.txt

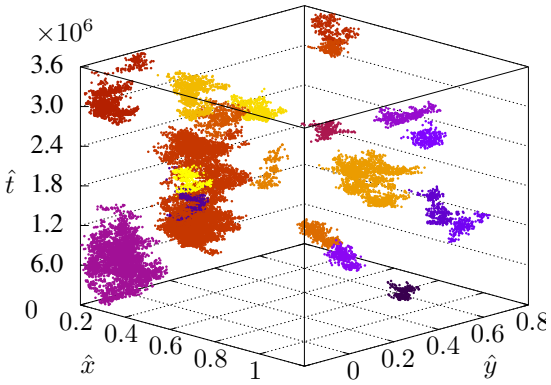

Fig. 2: Top-20 clusters in $\mathcal{D}_{\text{syn}}$. Different clusters are color-coded and the parameters under this simulation are $T = 3.6 \times 10^6$ seconds, $\eta = 8 \times 10^{-5}$, $\phi = 0.99$, $\omega = 10^{-4}$ and $\sigma = 10^{-2}$.

$\mathcal{D}_{\text{Hry}}$. In Figure 3, we present the flooding events and color-code the timestamps. The color bar range starts at 00:00:00 on 08/23/2017, and we can also observe the pattern of disaster-related events, where the events are reported mostly in urban regions and mostly clustered in space and time.

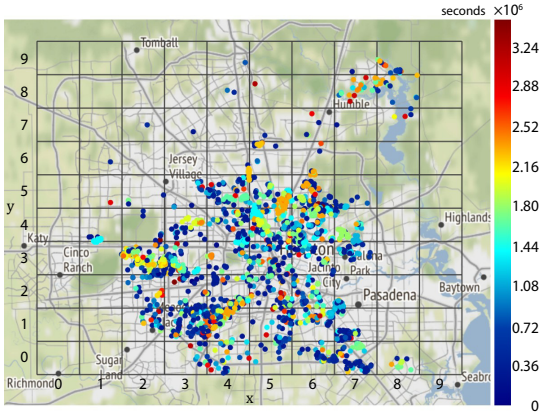

Fig. 3: Flooding Events in Houston $\mathcal{D}_{\text{Hry}}$. Each event is scaled and color-coded by its timestamps, and x-axis and y-axis represents the grid cell ID.

3) Reports of IED attack in Iraq ($\mathcal{D}_{IED}$):: In addition to rescue mission during natural disaster, we also test our framework on a spatio-temporal bomb detection and removal task that also requires exploration in the under-sampled areas. In specific, we compare our model with baselines on the reports of improvised explosive device attack (IED) in Iraq. In total, we focus on the 28,593 incidents mainly in Iraq region between 02/04 and 02/24 in 2009, and denote them as $\mathcal{D}_{\text{IED}}$.

B. Experimental Protocol

Given a spatial domain, we first partition the range of longitude and latitude of $\mathcal{D}_{\text{Syn}}$ and $\mathcal{D}_{\text{Hry}}$ evenly into 10 disjoint

intervals (i.e., $X = 10$ and $Y = 10$). Thus, there are 100 grid cells and every event of interest can be mapped to a unique grid cell. (We let $X = 20$ and $Y = 20$ as for $\mathcal{D}_{\text{IED}}$ due to its larger domain.) Next, we validate the five competing models on both the synthetic datasets ($\mathcal{D}_{\text{syn}}$) and the real-world dataset ($\mathcal{D}_{\text{Hry}}$ and $\mathcal{D}_{\text{IED}}$) by using the "walk-forward" validation approach. For example, for the 311 service request dataset $\mathcal{D}_{\text{Hry}}$, we discretize time starting from 08/23/2017 00:00:00 into intervals of 20 hours ($W = 72,000$ seconds). In each window of time we select $N = 10$ cells to visit. [‡] We then calculate the reward of the events that happen during our visits in the selected cells until the next visit. We also add these events to the historical training data set for updating the model in the next round, whereas events occurring in cells not visited are unseen by the model in the next round. We then slide the window and train the models on the historical observations up to the end of the previous observation window to make recommendations for the next visit. We then record the events that happen between 08/23/2017 20:00:00 and 08/24/2017 16:00:00. We repeat this process until the final date of 10/02/2020 24:00:00.For each grid cell, we sample 50 sets of parameters from the posterior distribution, i.e., $L = 50$. Since the selected cells at the beginning may result in different decisions and performances in the whole MAB process, for every parameter in all the models, we run MAB processes for 30 times with different initial visited cells, and we report the average of each evaluation score. Also, all parameters of the models are studied through an extensive grid search, and the best performances are reported for the model comparison. For the sake of reproducibility, all datasets and the source code are made publicly available in an anonymized repository [§].

C. Evaluation Metrics

We first measure the performance of competing models by the cumulative reward, that is, the number of the observed events captured in visited cells. To compare the performances between different datasets in $\mathcal{D}_{\text{syn}}$, we then normalize the total reward by the most optimal reward, i.e., the maximum rewards if the choice of levers are optimal, and we denote it as $\overline{\texttt{reward}}$. At each visit, models generate a short ranked list for the next visit. Based on the ranked lists, we can also evaluate the models through different ranking and recommendation metrics. One popular metric to evaluate the ranking quality is the normalized discounted cumulative gain (NDCG) [15]. We then calculate the NDCG at N for each visit, where N is the number of visited cells. The relevance value (i.e., gain) at cell a and visit v is then defined as the number of events, i.e., $|\mathcal{T}_v^a|$. Finally, we take the average across all the visits and denote it as NDCG.

From the recommendation point of view, we are interested in how many cells recommended by the models would actually contain events during our visit. We first consider that a cell is relevant if there are one or more events during the visit. We then evaluate such recommendation quality through the modified reciprocal hit rank [22], denoted as mRHR for evaluation. Modified reciprocal hit rank is a modified version of average reciprocal hit rank (ARHR), which is feasible for ranked recommendation evaluations where there are multiples relevant items (i.e., relevant cells). It can be calculated as follows:

$$\texttt{mRHR} = \frac{1}{\mid \boldsymbol{g} \mid} \sum_{i=1}^{N} \frac{\texttt{h}_i}{\texttt{r}_i}, \quad \text{where } \texttt{h}_i = \begin{cases} 1 & \text{if } a_i \in \boldsymbol{g} \\ 0 & \text{if } a_i, \notin \boldsymbol{g} \end{cases}, \quad \texttt{r}_i = \begin{cases} \texttt{r}_{i-1} & \text{if } \texttt{h}_{i-1} = 1 \\ \texttt{r}_{i-1}+1 & \text{if } \texttt{h}_{i-1} = 0, \end{cases} \tag{7}$$

where $\boldsymbol{g}$ is a list of relevant cells; $a_i \in \boldsymbol{a}$; h and r represent hit and rank, respectively; and each hit is rewarded based on its position in the ranked list. Last, we evaluate the models on F1 score [14], which is simply the harmonic mean between recall and precision of the relevant cells. Note that we omit the recall performance since it can be considered as the reward normalized by the total number of events which is known in our MAB process.

TABLE I: Best Performance on $\mathcal{D}_{\text{Hry}}$

Model	$\overline{\texttt{reward}}$	NDCG	mRHR	f1
ϵGdy	0.302 ± .059	0.309 ± .116	0.169 ± .050	0.277 ± .050
UCB1	0.393 ± .063	0.399 ± .058	0.213 ± .028	0.357 ± .036
$\text{UCB1}_{\text{HpSp}}$	0.426 ± .047	0.438 ± .028	0.243 ± .017	0.385 ± .015
SpUCB	0.435 ± .076	0.411 ± .070	0.210 ± .041	0.365 ± .025
HpSpUCB	**0.511** ± .072	**0.510** ± .055	**0.272** ± .011	**0.413** ± .013

D. Experimental Results

1) Performances on the synthetic datasets $\mathcal{D}_{syn}$: : We compare the performance of our model HpSpUCB against competitive baseline methods, UCB1 and SpUCB, in terms of $\overline{\texttt{reward}}$ when applied to synthetic datasets $\mathcal{D}_{\text{syn}}$ with different spatio-temporal patterns. The results of $\overline{\texttt{reward}}$ are presented in Figure 4. Figure 4 demonstrates the $\overline{\texttt{reward}}$ under different ω, while ϕ and σ while fixing the other parameters. Our HpSpUCB outperforms the other baselines by a large margin, with the exception of when the process is approximately stationary over moderate time scales. This occurs when ϕ or ω are too small or too large relative to the time scale of a visit, and in this scenario the Hawkes process loses its advantage over stationary models.

2) Performances on Houston 311 Service Requests $\mathcal{D}_{Hry}$:: Table I presents the best performance and its standard deviation of Houston 311 call dataset $\mathcal{D}_{\text{Hry}}$ according to each evaluation metric. In general, our model HpSpUCB outperforms all of the other baselines in every metric that we evaluate. In particular, by adding an event intensity tracking mechanism in the decision-making, the performance of HpSpUCB is better than SpUCB both on reward optimization and high-risk cell recommendation. In terms of $\overline{\text{reward}}$, the proposed HpSpUCB outperforms the second-best model, SpUCB, by 17.47% while it also surpasses SpUCB in NDCG by 24.09% from the ranking perspective. From the high-risk cell retrieval point of view, HpSpUCB consistently outperforms SpUCB in mRHR and f1

[‡] We select 5 cells ($N = 5$) to visit for a duration of 5 hours ($W = 18,000$) for synthetic datasets $\mathcal{D}_{\text{syn}}$ and we select 10 cells ($N = 10$) to visit for a duration of 40 days for $\mathcal{D}_{\text{IED}}$ due to its longer time frame.

[§] https://anonymous.4open.science/r/475a5b4d-9521-4c47-8bcb-94a5b2c1cae0/

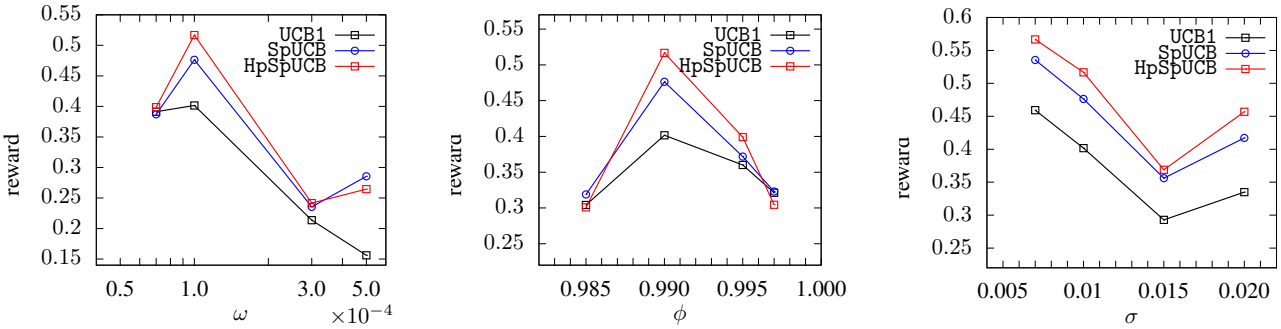

Fig. 4: Performance on $\overline{\texttt{reward}}$ on $\mathcal{D}_{\text{syn}}$ under different simulation scenarios with various of ω, ϕ and σ. We run simulations by changing only one of the parameters at a time and the other parameters, i.g., ω, ϕ and σ, are fixed at $10^{-4}, 0.99$ and 10^{-2}, respectively.

by 29.52% and 13.15%, respectively. These improvements in accuracy illustrate HpSpUCB's ability to recall events through event intensity tracking and provide better recommendations on the high-risk cells. By combining the method with the existing algorithm, stationary patterns of events are also taken into consideration and the combined model strikes a good balance between the $\mathcal{HP}$ component and the other UCB component.

Compared to UCB1, $\texttt{UCB1}_{\texttt{HpSp}}$ contains the proposed $\mathcal{HP}$ component. We can see from Table I that $\texttt{UCB1}_{\texttt{HpSp}}$ consistently outperforms UCB1 in all of the evaluation metrics. These results suggest that our proposed $\mathcal{HP}$ component is out-performing traditional stationary MAB algorithms like UCB1 by tracking the space-time dynamic reward distribution. Moreover, HpSpUCB outperforms SpUCB significantly with p-values 1.9×10^{-4}, 9.63×10^{-8}, 6.13×10^{-11}, and 3.79×10^{-13} across $\overline{\texttt{reward}}$, NDCG, mRHR and f1 through two tailed t-test. HpSpUCB also outperforms $\texttt{UCB1}_{\texttt{HpSp}}$ significantly with p-values 1.22×10^{-6}, 3.07×10^{-8}, 1.11×10^{-10}, 1.76×10^{-10} across these four evaluation metrics. In Figure 5 and Figure

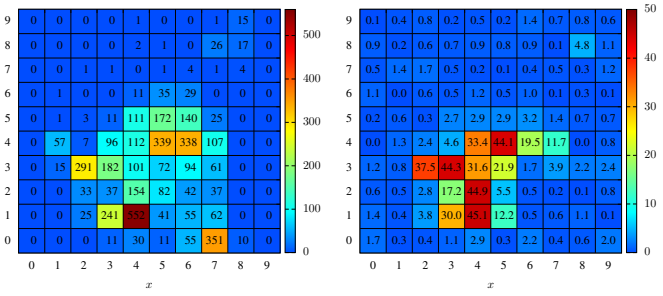

Fig. 5: Number of events

Fig. 6: Number of visits

6, we compare the number of flooding events and the average number of total visits for each grid cell in $\mathcal{D}_{\text{Hry}}$ among the first 10 MAB simulations from the best $\overline{\texttt{reward}}$ in our model HpSpUCB. We can see that the number of flooding events in the cells is highly correlated to the average number of visits at the end of the MAB process. This suggests that after the trial of exploration, eventually, our HpSpUCB will learn those cells that are most susceptible to flooding and focus on these in terms of exploitation. Figure 7 is the snapshot of the flooding

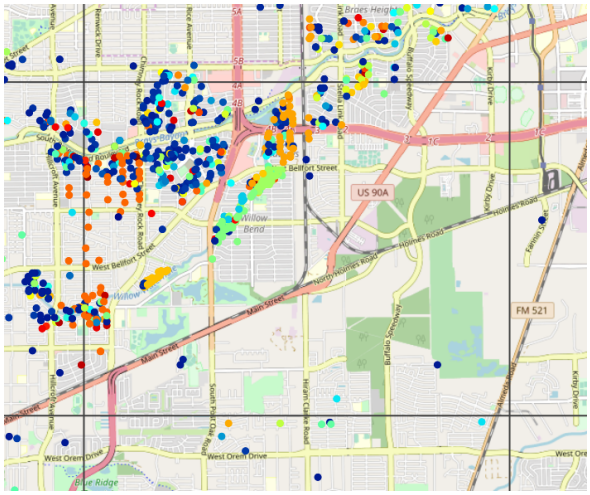

Fig. 7: Map on (4,1)

map in cell (4,1), that HpSpUCB gives the highest ranking on average. It is located by the watershed of The Brays Bayou, a slow-moving river which is notorious for its flooding history in Houston, Texas. This also indicates that HpSpUCB can identify hotspot areas for further investigation.

3) Performances on IED Attack Reports $\mathcal{D}_{IED}$:: In table II, we present the best performance and standard deviations of IED attack dataset $\mathcal{D}_{\text{IED}}$. Overall, the proposed HpSpUCB consistently outperform all the other baseline methods in all four metrics while the competitive baseline SpUCB comes in second. In terms of $\overline{\text{reward}}$ and NDCG, HpSpUCB improves 9.89% and 10.08% compared to SpUCB, respectively by a large margin. On the other hand, compared to $\overline{\text{reward}}$ and NDCG, our HpSpUCB only has slight improvements on mRHR and f1 (i.e., 4.23% and 2.04%, respectively) over SpUCB. This may be due to the fact that in $\mathcal{D}_{\text{IED}}$, there is a larger amount of grid cells and the bomb incidents are more evenly spread, which could cause the lack of quality on the recommendation of high-risk cells. Our HpSpUCB performs better than $\texttt{UCB1}_{\texttt{HpSp}}$ significantly with p-values 5.34×10^{-22}, 3.74×10^{-22}, 8.11×10^{-36}, 3.56×10^{-37} across these four evaluation metrics.

V. Conclusion

We introduced a novel framework HpSpUCB that integrates Bayesian Hawkes processes ($\mathcal{HP}$) with a spatial multi-armed

TABLE II: Best Performance on $\mathcal{D}_{\text{IED}}$

Model	$\overline{\texttt{reward}}$	NDCG	mRHR	f1
ϵ-Gdy	0.174 ± .055	0.186 ± .079	0.047 ± .008	0.107 ± .015
UCB1	0.122 ± .016	0.169 ± .028	0.028 ± .004	0.082 ± .004
$\text{UCB1}_{\text{HpSp}}$	0.070 ± .006	0.099 ± .011	0.026 ± .002	0.075 ± .004
SpUCB	0.455 ± .152	0.446 ± .142	0.071 ± .010	0.147 ± .013
HpSpUCB	**0.500** ± .154	**0.491** ± .139	**0.074** ± .009	**0.150** ± .013

bandit (MAB) algorithm to forecast spatio-temporal events and detect hotspots where disaster search and rescue efforts may be directed. In particular, the model forecasts synthetic events between each visit to a geographical area to infer the intensity in the gap between between visits. An upper confidence bound on the estimated intensity is then built for dynamic event tracking. We then apply a Gaussian filter to incorporate the spatial relationships between grid cells. We compared our HpSpUCB against competitive baselines through extensive experiments. In simulated synthetic datasets with space-time clustering, our HpSpUCB improves upon existing stationary spatial MAB algorithms. In the case of Houston 311 service requests during hurricane Harvey and IED attack reports in Iraq, HpSpUCB outperforms the baseline models considered in terms of a variety of metrics including total reward and ranking quality. Overall, with the $\mathcal{HP}$ component, we can enhance the performance of MAB algorithms. In the future, more contextual information may be used to further improve point process MAB algorithms. Furthermore, other types of point processes (log-Gaussian Cox processes, self-avoiding processes, etc.) may be combined with multi-armed bandits to solve other types of applications.

VI. Acknowledgements

This research was supported by NSF grants SCC-2125319 and ATD-2124313, and AFOSR MURI grant FA9550-22-1-0380.

References

[1] V. Avadhanula, R. Colini Baldeschi, S. Leonardi, K. A. Sankararaman, and O. Schrijvers, "Stochastic bandits for multi-platform budget optimization in online advertising," in *Proceedings of the Web Conference 2021*, 2021, pp. 2805–2817.

[2] E. Bacry, M. Bompaire, S. Gaïffas, and S. Poulsen, "Tick: a python library for statistical learning, with a particular emphasis on time-dependent modelling," *arXiv preprint arXiv:1707.03003*, 2017.

[3] E. Bacry, I. Mastromatteo, and J.-F. Muzy, "Hawkes processes in finance," *Market Microstructure and Liquidity*, vol. 1, no. 01, p. 1550005, 2015.

[4] F. Cheong and C. Cheong, "Social media data mining: A social network analysis of tweets during the 2010-2011 australian floods." *PACIS*, vol. 11, pp. 46–46, 2011.

[5] W.-H. Chiang, B. Yuan, H. Li, B. Wang, A. Bertozzi, J. Carter, B. Ray, and G. Mohler, "Sos-ew: System for overdose spike early warning using drug mover's distance-based hawkes processes," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 538–554.

[6] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 208–214.

[7] A. Durand, C. Achilleos, D. Iacovides, K. Strati, G. D. Mitsis, and J. Pineau, "Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis," in *Machine Learning for Healthcare Conference*, 2018, pp. 67–82.

[8] D. Eckles and M. Kaptein, "Thompson sampling with the online bootstrap," *arXiv preprint arXiv:1410.4009*, 2014.

[9] E. W. Fox, F. P. Schoenberg, J. S. Gordon *et al.*, "Spatially inhomogeneous background rate estimators and uncertainty quantification for nonparametric hawkes point process models of earthquake occurrences," *The Annals of Applied Statistics*, vol. 10, no. 3, pp. 1725–1756, 2016.

[10] A. Gopalan, S. Mannor, and Y. Mansour, "Thompson sampling for complex online problems," in *International Conference on Machine Learning*, 2014, pp. 100–108.

[11] S. Goswami, S. Chakraborty, S. Ghosh, A. Chakrabarti, and B. Chakraborty, "A review on application of data mining techniques to combat natural disasters," *Ain Shams Engineering Journal*, vol. 9, no. 3, pp. 365–378, 2018.

[12] D. Guo, S. I. Ktena, P. K. Myana, F. Huszar, W. Shi, A. Tejani, M. Kneier, and S. Das, "Deep bayesian bandits: Exploring in online personalized recommendations," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 456–461.

[13] N. Gupta, O.-C. Granmo, and A. Agrawala, "Thompson sampling for dynamic multi-armed bandits," in *2011 10th International Conference on Machine Learning and Applications and Workshops*, vol. 1. IEEE, 2011, pp. 484–489.

[14] G. Hripcsak and A. S. Rothschild, "Agreement, the f-measure, and reliability in information retrieval," *Journal of the American Medical Informatics Association*, vol. 12, no. 3, pp. 296–298, 2005.

[15] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.

[16] A. Krause and C. S. Ong, "Contextual gaussian process bandit optimization," in *Advances in neural information processing systems*, 2011, pp. 2447–2455.

[17] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," *arXiv preprint arXiv:1402.6028*, 2014.

[18] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 661–670.

[19] B. Merz, H. Kreibich, and U. Lall, "Multi-variate flood damage assessment: a tree-based data-mining approach," *Natural Hazards and Earth System Sciences (NHESS)*, vol. 13, no. 1, pp. 53–64, 2013.

[20] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, "Self-exciting point process modeling of crime," *Journal of the American Statistical Association*, vol. 106, no. 493, pp. 100–108, 2011.

[21] J. Møller and J. G. Rasmussen, "Perfect simulation of hawkes processes," *Advances in applied probability*, vol. 37, no. 3, pp. 629–646, 2005.

[22] S. Peker and A. Kocyigit, "mrhr: a modified reciprocal hit rank metric for ranking evaluation of multiple preferences in top-n recommender systems," in *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*. Springer, 2016, pp. 320–329.

[23] C. Rasmussen and C. Williams, "Gaussian processes for machine learning the mit press," 2006.

[24] J. G. Rasmussen, "Bayesian inference for hawkes processes," *Methodology and Computing in Applied Probability*, vol. 15, no. 3, pp. 623–642, 2013.

[25] G. O. Roberts, A. Gelman, W. R. Gilks *et al.*, "Weak convergence and optimal scaling of random walk metropolis algorithms," *The annals of applied probability*, vol. 7, no. 1, pp. 110–120, 1997.

[26] A. Semenov, M. Rysz, G. Pandey, and G. Xu, "Diversity in news recommendations using contextual bandits," *Expert Systems with Applications*, vol. 195, p. 116478, 2022.

[27] W. R. Smith, K. K. Stephens, B. Robertson, J. Li, and D. Murthy, "Social media in citizen-led disaster response: Rescuer roles, coordination challenges, and untapped potential," in *Proceedings of the... International ISCRAM Conference*, 2018.

[28] M. S. Tehrany, B. Pradhan, and M. N. Jebur, "Spatial prediction of flood susceptible areas using rule based decision tree (dt) and a novel ensemble bivariate and multivariate statistical models in gis," *Journal of Hydrology*, vol. 504, pp. 69–79, 2013.

[29] M. Tokic and G. Palm, "Value-difference based exploration: adaptive control between epsilon-greedy and softmax," in *Annual Conference on Artificial Intelligence*. Springer, 2011, pp. 335–346.

[30] L. Tran-Thanh, A. Chapman, E. M. de Cote, A. Rogers, and N. R. Jennings, "Epsilon–first policies for budget–limited multi-armed bandits," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[31] C. M. Wu, E. Schulz, M. Speekenbrink, J. D. Nelson, and B. Meder, "Mapping the unknown: The spatially correlated multi-armed bandit," *bioRxiv*, p. 106286, 2017.