Inverse Reinforcement Learning for Permanental Process

Sellier, Jeremy
jeremy.sellier.18@ucl.ac.uk

May 2, 2021

1 Preliminaries

1.1 The Poisson Process

A Poisson process is a random sequence of events occurring on a continuous domain $\mathcal{X}$. The general approach to infer point process is to estimate its intensity function $\lambda(x) : \mathcal{X} \mapsto \mathbb{R}^+$. The intensity $\lambda(x)$ can be interpreted heuristically as the instantaneous probability of occurrence of an event around a location x, i.e.

$$\lambda(x) := \lim_{\mu(dx)\to+0} \frac{E\left[N(dx)\right]}{\mu(dx)}$$

where dx is a small region around x with measure $\mu(dx)$ and $N(A)$ is the random number of points within a sub-region $A \subset \mathcal{X}$. Then given some observations $\mathbf{x} = \{x_i\}_{i=1}^N$ we can define the likelihood of data as

$$p(\mathbf{x}|\lambda(\cdot)) = \exp\left(-\int_{\mathcal{X}} \lambda(x)\, d\mu(x)\right) \prod_{i=1}^{N} \lambda(x_i). \tag{1}$$

1.2 Gaussian Cox Process

A Cox process, also known as a doubly stochastic Poisson process is a point process which is a generalization of a Poisson process where the intensity is itself a stochastic process. A Gaussian Process Cox process is obtained by placing a GP prior on λ. we also need to add a deterministic "link" function $\ell : \mathbb{R} \mapsto \mathbb{R}^+$ to that ensure the intensity function remains non-negative, leading to

$$\begin{aligned} &\lambda(\cdot) := \ell(f(\cdot)) \\ &f \sim \mathcal{GP}(0, K(x, x')). \end{aligned} \tag{2}$$

We have the posteriors

$$p(\mathbf{x}|\mathbf{f}) = \exp\left(-\int_{\mathcal{X}} \ell(f(x))\, dx\right) \prod_{i=1}^{N} \ell(f(x_i)) \tag{3}$$

$$p(\mathbf{f}|\mathbf{x}) = \frac{\exp\big(-\int_{\mathcal{X}} \ell(f(x))\, dx\big)\big[\prod_{i=1}^{N} \ell(f(x_i))\big]\mathcal{GP}(f)}{\int \exp\big(-\int_{\mathcal{X}} \ell(f(x))\, dx\big)\big[\prod_{i=1}^{N} \ell(f(x_i))\big]\mathcal{GP}(f)\, df}$$

which are often described as "doubly-intractable" because of the integral of the intensity function and the nested integral in the denominator, which typically cannot be calculated explicitly.

2 Permanental Process

We now consider the general case spatial where $\mathcal{X} = \mathbb{R}^2$ and restrict the domain of observations to a bounded window $\mathcal{S} \subset \mathbb{R}^2$. The absence of a closed form for the integral in (5) makes the fitting of Cox point process models to point pattern data difficult. Fitting even simple Cox processes has typically used MCMC methodology, and has been extremely computationally expensive.

However, a more flexible class of Gaussian Cox process, called *permanental process*, provides exception in that analytical expression for their density are available. They are obtained by defining the intensity in (5) as a square of Gaussian processes i.e setting $l(\cdot) = (\cdot)^2$. We expose here some brief results regarding permanental process taken from McCullagh and Moller [7].

Integral expression via Mercer theorem Let's define for the Gaussian process covariance function a positive definite kernel function $k : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$. If the domain $\mathcal{S}$ is compact and k is continuous, one define an integral operator $T_k : L(\mathcal{S}, \mu)^2 \to L(\mathcal{S}, \mu)^2$ given by $T_k f = \int_{\mathcal{S}} k(x, \cdot) f(x) d\mu(x)$ which is self-adjoint, positive and compact. Thus there exists countable orthonormal set of eigenfunction $\{\phi_i\}_{i=1}^{\infty}$ of T_k and corresponding eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$ such that $\{|\lambda_i|\}_{i=1}$ is a countable decreasing sequence and $\lim_{i\to\infty 0} \lambda_i = 0$. This leads to the Mercer representation of the kernel function

$$k_\theta(x, x') = \sum_{i=1}^{\infty} \lambda_i \Phi_i(x) \Phi_i(x') \quad \text{for } x, x' \in \mathcal{S} \tag{4}$$

with uniform convergence in $\mathcal{S} \times \mathcal{S}$. In our case, we will define μ to be the Lebesgue measure i.e $\mu(dx) = dx$. Then similarly to [7], [10] and [6], we can take reformulate $f(x) \sim GP(0, k(x, x'))$ to an equivalent to an equivalent linear form $f(x) = \sum_{i=0}^{\infty} w_i \Phi_i(x)$ where w_i are independent $\mathcal{N}(0, \lambda_i)$ distributed random variables. Indeed one can verify that

$$\mathrm{Cov}(f(x), f(x')) = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} E\left[w_i w_j\right] \Phi_i(x) \Phi_j(x') = \sum_{i=0}^{\infty} \lambda_i \Phi_i(x) \Phi_i(x') = k(x, x')$$

We can now express the integral of the intensity as follow

$$\begin{aligned}
\int_{\mathcal{S}} f(x)^2 \, dx &= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \int_{\mathcal{S}} w_i w_j \Phi_i(x) \Phi_j(x) dx \\
&= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} w_i w_j \langle \Phi_i(x), \Phi_j(x) \rangle_{L(\mathcal{S})^2} \\
&= \sum_{i=0}^{\infty} w_i^2
\end{aligned}$$

Likelihood expression The conditional likelihood now becomes

$$p(\mathbf{x}|\mathbf{w}) = \exp\big(-\sum_{k=0}^{\infty} w_k^2\big) \left[\prod_{i=1}^{N} \big(\sum_{j=0}^{\infty} w_j \Phi_j(x_i)\big)^2\right] \tag{5}$$

From this, the authors [7] derived an analytical expression of the marginal likelihood based on the permanent of the covariance function k

$$\begin{aligned}
p(\mathbf{x}) &= \int p(\mathbf{x}|\mathbf{w}) p(\mathbf{w}) d\mathbf{w} \\
&= E[p(\mathbf{x}|\mathbf{w})] \\
&= \mathrm{per}_\alpha[\tilde{k}](\mathbf{x}) \prod_{j=0}^{\infty} (1 + \lambda_j)^{-\frac{1}{2}}
\end{aligned}$$

where $\tilde{k}$ is the kernel defined via the mercer expansion $\tilde{k}(x,x') = \sum_{i=0}^{\infty}(\lambda_i/(1+\lambda_i))\Phi_i(x)\Phi_i(x')$ and $\text{per}_\alpha[\cdot]$ denotes the α weighted permanent i.e.

$$\text{per}_\alpha[\tilde{k}] = \sum_\sigma \alpha^{\#\sigma}\tilde{k}(x_1, x_{\sigma(1)})\dots\tilde{k}(x_N, x_{\sigma(N)})$$

where σ is a permutation of $\{1,\dots,N\}$ and $\#\sigma$ is the number of cycles of the permutation σ. The computation of the permanent terms is difficult in practise. Valient [9] has shown that there is no available deterministic polynomial-time expression for general matrices.

Interestingly, the expectation of the conditional log-likelihood can also be solved analytically, which is consistent with the previous results found in [3] for the square link function (see derivation in appendix **??**).

$$E[\log p(\mathbf{x}|\mathbf{w})] = \sum_{i=1}^{N}\log k_\theta(x_i,x_i) - N(C+\log(2)) - \sum_{k=0}^{\infty}\lambda_k$$

3 Inverse Reinforcement Learning framework

Li et al. in [12] and [11] proposes a reinforcement learning framework to learn point process models where the reward function is this time obtained via inverse reinforcement learning (IRL) from the observed sequence. To avoid the expensive computation of an IRL the function class of reward is chosen to be the unit ball in RKHS. This reduces the IRL setting to a more simple minimization problem.

RL setting We defined two spatial point process over $\mathcal{S}$: an *expert* observed point process $\mathbf{x} = \{x_1, ..., x_{N_E}\} \sim \pi_E$ and a point process sampled from the agent policy $\mathbf{t} = \{t_1, ..., t_{N_\theta}\} \sim \pi_\theta$. A reward function $r^*(\cdot)$ is assumed to be received over $\mathcal{S}$ and depends on the trajectories of $\mathbf{t}$ and $\mathbf{x}$. The goal is to find an optimal policy π_θ^* that maximizes the expected cumulative reward function, i.e.

$$\pi_\theta^*(\cdot) = \arg\max_{\pi_\theta\in\mathcal{G}} J(\pi_\theta) := \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r^*(t_i)\right] \tag{6}$$

where $\mathcal{G}$ is a family of all candidates policies.

IRL In our case the expert sequence of event can be observed but the optimal reward function $r^*(\cdot)$ is unknown. We can be recovered from an inverse reinforcement learning setting. The optimal reward function is set to be the one that maximize the cumulative reward discrepancies $R^*(T)$ between the expert policy and the best possible agent policy in $\mathcal{G}$.

$$\begin{aligned} R^*(T) &= \max_{r\in\mathcal{F}}\left(\mathbb{E}_{\mathbf{x}\sim\pi_{\mathbb{E}}}\left[\sum_{i=1}^{N_E} r(x_i)\right] - \max_{\pi_\theta\in\mathcal{G}}\mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r(t_i)\right]\right) \\ &= \max_{r\in\mathcal{F}}\min_{\pi_\theta\in\mathcal{G}}\left(\mathbb{E}_{\mathbf{x}\sim\pi_{\mathbb{E}}}\left[\sum_{i=1}^{N_E} r(x_i)\right] - \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r(t_i)\right]\right) \end{aligned} \tag{7}$$

where $\mathcal{F}$ is a family of reward functions. Notice that once the optimal reward function r^* that attains R^* is fixed, the min part is attained by π_θ^*.

Reward function class restriction and MMD We now put some restriction to the function class $\mathcal{F}$ to be able to learn the oprimal policy π_θ^*. One possibility proposed by [12] and [11], is to restrict $\mathcal{F}$ to be a unit ball in a RKHS $\mathcal{H}$ i.e. $\mathcal{F} := \{r\in\mathcal{H}, ||r||_{\mathcal{H}} \le 1\}$. The space is dense enough for being an acceptable restriction. Li et al. [12] showed that the expensive minimax problem can then be transformed into a simple minimization problem over π_θ (see theorem 2) that is easier and stable to solve. This result shares strong

similarities with the Maximum Mean Discrepancy (MMD) derivation proposed by Gretton et al [2]. First, the existence of a point process *mean-embedding* to the RKHS $\mathcal{H}$ is established (see proposition 1). The problem is then transformed to a minimax problem and the nested max is solved analytically, to finally obtained a min only problem.

We first define below the notion of embedding a point process measure into a RKHS. It bears direct similitude to the embedding of a probability measure into a RKHS as defined in [2].

Proposition 1. *Let's assume a regular point process N_θ on a compact space $\mathcal{S}$. If $k(.,.)$ the reproducible kernel of a RKHS $\mathcal{H}$ is measurable and $\mathbb{E}_{\mathbf{x}\sim\pi_\theta}\left[\int_{\mathcal{S}} k(t,t)^{\frac{1}{2}}\, dN_\theta(t)\right] \leq \infty$, there exist a mean embedding $\mu_{\pi_\theta} \in \mathcal{H}$ such that :*

$$\mathbb{E}_{\eta\sim\pi_\theta}\left[\int_{\mathcal{S}} r(t)\, dN_\theta(t)\right] = \langle r, \mu_{\pi_\theta}\rangle, \quad \forall r \in \mathcal{H}$$

Moreover,

$$\mu_{\pi_\theta}(\cdot) := \mathbb{E}_{\eta\sim\pi_\theta}\left[\int_{\mathcal{S}} k(t,.)\, dN_\theta(t)\right]$$

Proof. see proof in appendix A. □

The problem (6) to then transformed into a simpler min problem over π_θ presented in theorem (2) below.

Proposition 2. *Let the family of reward function be restricted to the unit ball of the RKHS $\mathcal{H}$ i.e $||\mathcal{H}||_{\mathcal{H}} \leq 1$. Then the optimal policy can be obtained from*

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

where

$$D(\pi_E, \pi_\theta, \mathcal{H}) := \mathbb{E}_{\mathbf{x}\sim\pi_{\mathbb{E}}}\left[\sum_{i=1}^{N_E} r^*(x_i)\right] - \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r^*(t_i)\right]$$

with an optimal reward function $r^*(\cdot) \propto \mu_{\pi_E}(\cdot) - \mu_{\pi_\theta}(\cdot)$.

Proof. see proof in appendix B. □

Finite Sample estimation We can also provide a sample approximation for μ_{π_θ} and μ_{π_E}. Given L trajectories of observed events and M trajectories of events generated by π_θ, we have

$$\mu_{\pi_E}(\cdot) = \mathbb{E}_{\mathbf{x}\sim\pi_{\mathbb{E}}}\left[\sum_{i=1}^{N_E} k(x_i,\cdot)\right] \approx \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{N_E^{(l)}} k(x_i^{(l)},\cdot) := \hat{\mu}_{\pi_E}(\cdot)$$

Then,

$$\hat{r}^*(t) \propto \hat{\mu}_{\pi_E}(t) - \hat{\mu}_{\pi_\theta}(t) = \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{N_E^{(l)}} k(x_i^{(l)},t) - \frac{1}{M}\sum_{l=1}^{M}\sum_{i=1}^{N_\theta^{(m)}} k(t_i^{(m)},t) \tag{8}$$

Learning via policy gradient We can finally learn the optimal policy π_θ from (14) using policy gradient. With the the likelihood ratio trick, we obtain

$$\begin{aligned}\nabla_\theta D(\pi_E, \pi_{\theta,\mathcal{H}}) &= -\nabla_\theta \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\left(\sum_{i=1}^{N_\theta} \hat{r}^*(t_i)\right)\right] \\ &= -\mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\left(\nabla_\theta \log p_{N_\theta}(\mathbf{t})\right)\left(\sum_{i=1}^{N_\theta} \hat{r}^*(t_i)\right)\right] \end{aligned} \tag{9}$$

where $p_n(\mathbf{t})$ denotes the local *janossy* density over $\mathcal{S}$ which is the density of having exactly n points $\{t_1, \ldots, t_n : t_1 < \ldots, < t_n\}$ in $\mathcal{S}$. Notice that by doing so, the authors implicitly assumed that once $\hat{r}^*(t)$ has been obtained by pre-sampling some trajectories of π_θ and π_E, it losses any dependency to θ. The overall procedure is given in algorithm (1).

Algorithm 1 IRL algorithm for Point process

1: **for** $t = 1, 2, \ldots\ldots$ **do**

2: Sample L trajectories of points $\{\epsilon^{(1)}, \ldots, \epsilon^{(L)}\}$ from agent policy π_E where $\epsilon^{(l)} = \{x_1^{(l)}, \ldots, x_{N^{(l)}}^{(l)}\}$

3: Sample M trajectories of points $\{\eta^{(1)}, \ldots, \eta^{(M)}\}$ from agent policy π_θ where $\eta^{(m)} = \{t_1^{(m)}, \ldots, t_{N^{(m)}}^{(m)}\}$

4: Estimate the policy gradient as

$$\nabla_\theta D(\pi_E, \pi_{\theta,\mathcal{H}}) = \frac{1}{M}\sum_{m=1}^{M} \nabla_\theta \log p_\theta(\eta^{(m)})\left(\sum_{i=1}^{N^{(m)}} \hat{r}^*(t_i^{(m)})\right)$$

where $\hat{r}^*(t_i^{(m)})$ can be estimated from the L trajectories of expert and $M-1$ other trajectories of agent as

$$\hat{r}^*(t^{(m)}) = \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{N^{(l)}} k(x_i^{(l)}, t^{(m)}) - \frac{1}{M-1}\sum_{m'=1, m'\neq m}^{M}\sum_{k=1}^{N^{(m')}} k(t_j^{(m')}, t^{(m)})$$

5: update parameter with SGD

$$\theta \leftarrow \theta + \alpha \nabla_\theta D(\pi_E, \pi_{\theta,\mathcal{H}})$$

6: **end for**

7: **return** θ

4 IRL for Permanental Point Process

In our permanental process case the family of candidate policies $\mathcal{G}$ is restricted to be the family of local *janossy* density over $\mathcal{S}$ with GP based intensity function as in (2). We are thus left with an optimization problem over the GP hyperparameters θ. The problem (7) becomes :

$$\hat{\theta} = \arg\max_{\substack{\theta \\ \text{s.t } \pi_\theta \in \mathcal{G}}} J(\pi_\theta).$$

4.1 Inference

Now we can learn the GP hyperparameters using policy gradient. Notice that in our case the log-likelihood $\log p_n(\mathbf{t})$ present in (9) is intractable. We thus need to the expectation term by reintroducing the Gaussian latent variable $\mathbf{z}$ previously defined. We obtain (see Appendix C),

$$\begin{aligned}\nabla_\theta J(\pi_\theta) &= \nabla_\theta \, \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r^*(t_i)\right] \\ &= \mathbb{E}_{(\mathbf{t},\mathbf{z})\sim\pi_\theta}\left[\nabla_\theta \log p_{N_\theta}(\mathbf{t}|\mathbf{z})\left(\sum_{i=1}^{N_\theta} r^*(t_i)\right)\right] \end{aligned} \tag{10}$$

where the last expectation is over the latent $\mathbf{z}$ and the point process conditioned on $\mathbf{z}$. The term $p_n(\mathbf{t}|\mathbf{z})$ denotes the local *janossy* density knowing $\mathbf{z}$, i.e. is equal to $\exp(-\mathbf{z}^\top \Lambda \mathbf{z}) \prod_{i=1}^n \left(\mathbf{z}^\top \Phi(t_i)\right)^2$. The optimal reward function $r^*(\cdot)$ can be estimated as in (9).

4.2 Approximate Sampling of Permanental process

Point processes are usually be sampled via *thinning* which is a variation of acceptation rejection proposed by Lewis and Shelder [4]. Adams et al. used in [1] a simple and unbiased extension of the thinning procedure for Gaussian Cox process (see Appendix D). As for standard thinning, it relies on the assumption that there exists an upper bound $\lambda^* > 0$ for the intensity function $\lambda(\cdot)$ of the process we wish to sample from. Unfortunately in our case, the square transformation leaves the intensity function unbounded. Thus we propose here different sampling approaches using thinning by first computing a bound approximation to the intensity. This bound approximation step relies on Bayesian optimization approach that we reminds here.

Bayesian optimization Bayesian optimization is a popular approach for finding the maximum of a non-convex function that is difficult to evaluate. It works by placing a GP prior in place of the objective function. The global optimum is approached by iteratively maximizing a so-called acquisition function, that balances the exploration and exploitation effect of the search. They usually use the predicted mean and predicted variance generated by the Gaussian process. The GP prior is updated at each iteration to form a more informative posterior distribution over the space of objective functions and concentrate the sampling as rapidly as possible to a near maxima. The performance of such an algorithm is largely affected by the choice of the acquisition function. We rely further on the GP-Upper Confidence Bound algorithm (GP-UCB) proposed by [5] (see appendix E).

Approximate Sampling 1 We want to sample the permanental process using the *thinning method* proposed by [1]. As seen in appendix (D), it requires to generate points from an inhomogeneous process with intensity λ^* and then evaluate the intensity function $\lambda(\cdot)$ at these points by sampling f from the GP prior. A first approach for estimating a λ^* is to first approximate a pseudo-upper and lower-bounds (f_*, f^*) to the GP sample using a Bayesian optimization step and then set $\lambda^* = \max(|f^*|, |f_*|)^2$. Finally, the thinning step is performed by sampling from the updated GP posterior (conditioned on the *pre-sampled* observations of the Bayesian optimization step). Notice that the Bayesian optimization step here differs from standard ones since there is no real function to observe. Observation are obtained by iteratively sampling from the GP of the intensity function itself. Conditioned on these pre-sampled observations, the probability of sampling further values superior to f^* or inferior to f_* is obviously non null. But we observe in practise that this

Bayesian sampling provides enough constraint to the GP posterior for f^* and f_* to be acceptable bound approximation to further sampling.

In more details, f^* and f_* can be estimated with an adapted version of the GP-UCB algorithm. Instead of solving two separate optimizations problem, we perform a single GP-UCB run where at each iterations we alternate a min and max search (see algo 2). By doing so, the min search also use the information contained in the sample generated for the max search and vice versa. Then f^* and f_* are determined by taking the *max* and *min* over all the pre-sampled observations. Combining the different pierces, we finally obtain a sampling algorithm (see 3).

Algorithm 2 The adapted GP-UCB algorithm for GP bound

Input: Input space $\mathcal{S}$, GP prior $(\mu_0, \sigma_0), \mathcal{O} = \{\emptyset\}$ by default
$a \leftarrow 1$
for $t = 1, 2 \dots, N$ **do**
  Choose $x_t = \arg\max_{x \in \mathcal{S}} a * (\mu_{t-1}(x) + \beta^{1/2}\sigma_{t-1}(x))$
  Sample $f(x_t)$ from the posterior $\mathcal{GP}(\mu_{t-1}(x), \sigma_{t-1}(x))$
  $\mathcal{O} \leftarrow \mathcal{O} \cup (x_t, f(x_t))$ ▷ Update the observations.
  $\mu_t, \sigma_t \leftarrow \mu_{t-1}, \sigma_{t-1}$ ▷ Perform Bayesian update to the GP mean and variance.
  $a \leftarrow (-1) * a$ ▷ Update a to alternatively switch the min/max in (3).
end for
$f^* = \min f(x_1), \dots f(x_N)$ and $f_* = \min f(x_1), \dots f(x_N)$
return $f^*, f_*, \mathcal{O}$

Algorithm 3 Permanental Process Approximate sampling 1

Input: Input space $\mathcal{S}$, GP prior (μ_0, σ_0)
$f^*, f_*, \mathcal{O} \leftarrow$ GP-UCB algo ▷ Return the bounds estimates from GP-UCB.
$\lambda^* \leftarrow \max(|f^*|, |f_*|)^2$
$K \sim Poisson(\lambda^* |\mathcal{S}|)$ ▷ Draw the number of points.
$\{s_i\}_{i=1}^K \sim Uniform(|\mathcal{S}|)$ ▷ Distribute the points uniformly in $|\mathcal{S}|$.
$\{f(s_i)\}_{i=1}^K \sim \mathcal{GP}(\mu, K, \mathcal{O})$ ▷ Sample from the posterior GP conditioned on $\mathcal{O}$.
$\mathcal{E} \leftarrow \{\emptyset\}$
for $i = 1, 2 \dots, K$ **do**
  $u_i \sim Uniform(0, 1)$ ▷ Draw uniform variable.
  if $u_i < f(s_i)^2/\lambda^*$ **then** ▷ Apply acceptance rule.
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{s_i\}$ ▷ Add s_i to accepted events.
end for
return $\mathcal{E}$

Approximate Sampling 2 We propose now another sampling method that exploits the augmented model form for permanental process described in part 2 and is thus more adapted for our inference expression in 4.2. In part 2 we saw that the intensity function is determined by $f(x) = \sum_{i=0}^{\infty} z_i \sqrt{\lambda_i} \Phi_i(x)$ where $\mathbf{z} \sim \mathcal{N}(0, I)$. Thus conditioned on the latent $\mathbf{z}$, $f(\cdot)$ becomes a deterministic function $\mathcal{S} \to \mathbb{R}$, i.e the intensity losses its stochasticity and the process becomes a simple inhomogeneous Poisson process. We can thus sample $\mathbf{z}$, find a bound on the obtained function f (using Bayesian optimization or other) and finally apply the thinning algorithm (see algo (4)). Notice that if we were to use Bayesian optimization, this time a different GP must be used for searching the bounds.

Algorithm 4 Permanental Process Approximate sampling 2

Input: Input space $\mathcal{S}$, GP prior (μ_0, σ_0)
Sample $\mathbf{z} \sim \mathcal{N}(0, I)$
Find the bounds (f_*, f^*) of $f(x) = \sum_{i=0}^{\infty} z_i \sqrt{\lambda_i} \Phi_i(x)$
$\lambda^* \leftarrow \max(|f^*|, |f_*|)^2$
$K \sim Poisson(\lambda^* |\mathcal{S}|)$ ▷ Draw the number of points.
$\{s_i\}_{i=1}^K \sim Uniform(|\mathcal{S}|)$ ▷ Distribute the points uniformly in $|\mathcal{S}|$.
$\{f(s_i)\}_{i=1}^K \sim \mathcal{GP}(\mu, K, \mathcal{O})$ ▷ Sample from the posterior GP conditioned on $\mathcal{O}$.
$\mathcal{E} \leftarrow \{\emptyset\}$
for $i = 1, 2 \ldots, K$ **do**
  $u_i \sim Uniform(0, 1)$ ▷ Draw uniform variable.
  if $u_i < f(s_i)^2 / \lambda^*$ **then** ▷ Apply acceptance rule.
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{s_i\}$ ▷ Add s_i to accepted events.
end for
return $\mathcal{E}$

5 Numerical approximation of the Mercer expansion via Nystrom method

Some kernels have an explicit Mercer decomposition when restricted to certain measure and domain. This is for example the case of the squared exponential kernel on $\mathbb{R}$ with the Gaussian measure. The eigenvector in 4 will be then orthogonal in $L(\mathbb{R}, \upsilon)^2$ where $\upsilon = \mathcal{N}(0, l^2 I)$. In our case, when we do not have a explicit representation with respect to $\mathcal{S}$ and the Lebesgue measure mercer, the eigenvector and eigenvalue of T_k can be approximated by the Nystrom method (see [8] Section 4.3). Both [6] and [10]) propose it in the context of permanental cox process where k is set to be the squared exponential kernel.

From the Mercer decomposition expression with the Lebesgue measure,

$$\lambda_k \Phi_i(x') = \int_{\mathcal{X}} k(x', x) \Phi_k(x) dx \approx \frac{|\mathcal{S}|}{n} \sum_{l=1}^{n} k(x', x_l^n) \Phi_i(x_l^n)$$

where the $\{x_l^n\}$ are sampled from a uniform distribution over $|\mathcal{S}|$. Setting $x' = x^n$ and $K_{n.n}$ the Gram matrix with i, j entries $k(x_i^n, x_j^n)$, leads to the eigenproblem

$$K_{n,n} \Phi_i^n = \lambda_k^n \Phi_i^n \quad \text{for } i = 1, \ldots, n$$

where Φ_i^n and λ_k^n are respectively the normalized eigenvectors and eigenvalues of the matrix $K_{n,n}$. The eigenfunctions and eigenvalues of k are thus approximated via Φ_i^n and λ_i^n as follows

$$\tilde{\lambda}_i = \frac{|\mathcal{S}|}{n} \lambda_i^n \qquad \text{for } i = 1, \ldots, n \tag{11}$$

$$\tilde{\Phi}_i(\cdot) = \frac{\sqrt{n/|\mathcal{S}|}}{\lambda_i^n} k(\cdot, \mathbf{x}^n) \Phi_i^n \qquad \text{for } i = 1, \ldots, n \tag{12}$$

The $\sqrt{n}$ factor arises from the differing normalizing factor between the eigenfunctions $\{\Phi_i(\cdot)\}$ of T_k and the eigenvectors $\{\Phi_i^n\}$ of $K_{n,n}$. Notice that this also produce for $n < N$ a low rank approximation of the kernel of the form

$$K_{x,x} \approx K_{x,n} \left(\sum_{i=1}^{n} \frac{1}{\lambda_i^n} (\Phi_i^n)(\Phi_i^n)^\top \right) K_{x,n} = K_{x,n} U^{(n)} \Lambda^{-1} {U^{(n)}}^\top K_{n,x} = K_{x,n} K_{n,n}^{-1} K_{n,x}$$

where $U^{(n)}$ denotes the matrix formed with the eigenvectors $\{\Phi_i^n\}$ as columns and Λ is the diagonal matrix with entries $\{\lambda_i^n\}$. The last equation holds by the EVD of $K_{n,n}$. Similarly the function f is now expressed

as

$$f(x) \approx \sum_{k=1}^{n} \tilde{w}_k \tilde{\Phi}_k(x) = k(x, \mathbf{x}^n) \sum_{i=1}^{n} \frac{z_k}{\sqrt{\lambda_k^n}} \Phi_k^n = k(x, \mathbf{x}^n) U \Lambda^{-\frac{1}{2}} \mathbf{z} \tag{13}$$

where $\mathbf{z} \sim \mathcal{N}(0, I)$.

Under the Nystrom method and from equation (5) the likelihood can be expressed as

$$p_n(\mathbf{x}|\mathbf{z}) = \exp(-\mathbf{z}^\top \Lambda \mathbf{z}) \prod_{i=1}^{n} \left(\tilde{f}(x_i) \right)^2$$

where $\mathbf{z} \sim \mathcal{N}(0, I)$ and $\Lambda = \text{diag}(\tilde{\lambda}_1, \ldots, \tilde{\lambda}_n)$ with $\tilde{\lambda}_i$ defined by equation () for all $i = 1, \ldots, n$ and from equation (13),

$$\tilde{f}(x) = k(x, \mathbf{x}^n) \sum_{i=1}^{n} \frac{z_k}{\sqrt{\lambda_k^n}} \Phi_k^n = k(x, \mathbf{x}^n) U \Lambda^{-\frac{1}{2}} \mathbf{z}.$$

Appendix A Point process *mean-embedding* into a RKHS

Proof of Proposition 1. The linear operator $T[r] := \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\int_{\mathcal{S}} r(t)\, dN_\theta(t)\right]$ verifies,

$$\begin{aligned}
|T[r]| &:= \left|\mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\int_{\mathcal{S}} r(t)\, dN_\theta(t)\right]\right| \\
&\leq \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\int_{\mathcal{S}} |r(t)|\ dN_\theta(t)\right] \\
&= \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\int_{\mathcal{S}} |\langle r, k(t,.)\rangle_{\mathcal{H}}|\ dN_\theta(t)\right] \qquad (\textit{Reproducing kernel property}) \\
&\leq \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\int_{\mathcal{S}} ||r||_{\mathcal{H}}\ ||k(t,.)||_{\mathcal{H}}\ dN_\theta(t)\right] \qquad (\textit{Cauchy-Schwarz inequality}) \\
&= \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\int_{\mathcal{S}} k(t,t)^{\frac{1}{2}}\ dN_\theta(t)\right]\ ||r||_{\mathcal{H}}
\end{aligned}$$

Hence, since $\mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\int_{\mathcal{S}} k(t,t)^{\frac{1}{2}}\ dN_\theta(t)\right] \leq \infty$ by assumption, $T[\cdot]$ is a bounded operator. Thus by the Riesz representer theorem there exist $\mu_{\pi_\theta} \in \mathcal{H}$ s.t $T[r] = \langle r, \mu_{\pi_\theta}\rangle_{\mathcal{H}},\ \forall r \in \mathcal{H}$. And if we set $r := k(t,.)$ then $T[k(t,.)] = \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\int_{\mathcal{S}} k(t,.)\, dN_\theta(t)\right] = \langle k(t,.), \mu_{\pi_\theta}\rangle_{\mathcal{H}} = \mu_{\pi_\theta}$. □

Verification of the Assumption for Stationary Kernel. Notice that for stationary kernel where $k(t,t') = k(t-t')$ and $k(0) = \gamma$, we have

$$\begin{aligned}
\mathbb{E}\left[\int_{\mathcal{S}} k(t,t)^{\frac{1}{2}}\ dN_\theta(t)\right] &= = \gamma^{\frac{1}{2}}\mathbb{E}\left[N_\theta(\mathcal{S})\right] \\
&= \gamma^{\frac{1}{2}}\mathbb{E}\left[\int_{\mathcal{S}} \lambda(t)dt\right] \\
&= \gamma^{\frac{1}{2}}\mathbb{E}\left[\int_{\mathcal{S}} f(t)^2\ dt\right] \\
&= \gamma^{\frac{1}{2}}\mathbb{E}\left[\sum_{i=0}^{\infty} w_i^2\right] \\
&= \gamma^{\frac{1}{2}}\sum_{i=0}^{\infty}\lambda_i \\
&= \gamma^{\frac{3}{2}}|\mathcal{S}| < \infty
\end{aligned}$$

□

Appendix B Proof of theorem (2)

We now provide a proof for the Theorem (2). We follow strictly Li et al. in [12]. From the proposition (1), we have that

$$\mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r(t_i)\right] = \mathbb{E}_{\eta\sim\pi_\theta}\left[\int_{\mathcal{S}} r(t)\; dN_\theta(t)\right] = \langle r, \mu_{\pi_\theta}\rangle$$

and similarly $\mathbb{E}_{\mathbf{x}\sim\pi_{\mathbb{E}}}\left[\sum_{i=1}^{N_E} r(t_i)\right] = \langle r, \mu_{\pi_E}\rangle$. Note that all Hilbert spaces are reflexive. Thus by Kakutani's Theorem, the closed unit balls $\{r \in \mathcal{H} : ||r||_{\mathcal{H}} \leq 1\}$ is weakly compact. Let's assume from now that $\mathcal{G}$ is also compact. And finally note that $\langle r, \mu_{\pi_E} - \mu_{\pi_\theta}\rangle$ is concave with respect to r and convex with respect to μ_{π_θ}. Then from the minimax theorem

$$\begin{aligned}
R^*(T) &= \max_{||r||_{\mathcal{H}}\leq 1} \min_{\pi_\theta\in\mathcal{G}} \left(\mathbb{E}_{\mathbf{x}\sim\pi_{\mathbb{E}}}\left[\sum_{i=1}^{N_E} r(x_i)\right] - \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r(t_i)\right]\right) \\
&= \max_{||r||_{\mathcal{H}}\leq 1} \min_{\pi_\theta\in\mathcal{G}} \langle r, \mu_{\pi_E} - \mu_{\pi_\theta}\rangle \\
&= \min_{\pi_\theta\in\mathcal{G}} \max_{||r||_{\mathcal{H}}\leq 1} \langle r, \mu_{\pi_E} - \mu_{\pi_\theta}\rangle \\
&= \min_{\pi_\theta\in\mathcal{G}} ||\mu_{\pi_E} - \mu_{\pi_\theta}||_{\mathcal{H}}
\end{aligned}$$

We define the optimal function r^* in the third line to be the RKHS function attaining the supremum $||\mu_{\pi_E} - \mu_{\pi_\theta}||_{\mathcal{H}}$ i.e. $r^*(\cdot) = \frac{\mu_{\pi_E}(\cdot) - \mu_{\pi_\theta}(\cdot)}{||\mu_{\pi_E} - \mu_{\pi_\theta}||_{\mathcal{H}}}$. The optimal policy π^*_θ is set to be the one attaining the cumulative reward discrepancies R^*. We are thus left with the equivalent problem

$$\pi^*_\theta(\cdot) = \arg\min_{\pi_\theta\in\mathcal{G}} \underbrace{\left(\mathbb{E}_{\mathbf{x}\sim\pi_{\mathbb{E}}}\left[\sum_{i=1}^{N_E} r^*(x_i)\right] - \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r^*(t_i)\right]\right)}_{:=D(\pi_E,\pi_\theta,\mathcal{H})} \tag{14}$$

$$\text{where} \quad r^*(\cdot) \propto \mu_{\pi_E}(\cdot) - \mu_{\pi_\theta}(\cdot)$$

Appendix C IRL Gradient for Permanental Process

$$
\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta \, \mathbb{E}_{\mathbf{t}\sim\pi_\theta}\left[\sum_{i=1}^{N_\theta} r^*(t_i)\right] \\
&= \nabla_\theta \sum_{n=0}^{\infty} \mathbb{E}\left[\sum_{i=1}^{N_T^\theta} r^*(t_i)\middle| N_\theta = n\right] Pr(N_\theta = n) \\
&= \nabla_\theta \sum_{n=0}^{\infty} \frac{1}{n!} \idotsint_{t_1\ldots,t_n\in\mathcal{S}} \left(\sum_{i=1}^{n} r^*(t_i)\right) p_n(t_1,\ldots,t_n)\, dt_1 \ldots\, dt_n \\
&= \nabla_\theta \sum_{n=0}^{\infty} \frac{1}{n!} \idotsint_{t_1\ldots,t_n\in\mathcal{S}} \left(\sum_{i=1}^{n} r^*(t_i)\right) \left(\int p_n(\mathbf{t}|\mathbf{z}) p_n(\mathbf{z}) d\mathbf{z}\right) dt_1 \ldots\, dt_n \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \idotsint_{t_1\ldots,t_n\in\mathcal{S}} \int \left(\sum_{i=1}^{n} r^*(t_i)\right) \nabla_\theta \log p_n(\mathbf{t}|\mathbf{z}) p_n(\mathbf{t},\mathbf{z})\, d\mathbf{z}\, dt_1 \ldots\, dt_n \\
&= \mathbb{E}_{(\mathbf{t},\mathbf{z})\sim\pi_\theta}\left[\nabla_\theta \log p_{N_\theta}(\mathbf{t}|\mathbf{z}) \left(\sum_{i=1}^{N_\theta} r^*(t_i)\right)\right]
\end{aligned}
$$

where $p_n(\mathbf{t})$ denotes the local *janossy* density over $\mathcal{S}$. Also in our Gaussian Cox Process case, $p_n(\mathbf{t}|\mathbf{z})$ denotes the local *janossy* density conditioned on the latent $\mathbf{z}$, i.e. from before $p_n(\mathbf{t}|\mathbf{z}) = \exp(-\mathbf{z}^\top \Lambda \mathbf{z}) \prod_{i=1}^{n} \left(\mathbf{z}^\top \Phi(t_i)\right)^2$.

Appendix D Thinning algorithm for Gaussian Cox process

The following algorithm has been proposed by Adams et al. [1]) to simulate Gaussian Cox process when the intensity function is bounded by a known $\lambda^* > 0$. In their paper, the authors worked on a Sigmoidal Gaussian Cox Process (i.e. with sigmoid link function for the intensity) that enables to fix a bound. The algorithm follows the step of a standard thinning algorithm for inhomogeneous point process except that the evaluation of the intensity is obtained by sampling from the GP prior.

Overall, we proceed by first simulating an homogeneous Poisson process with intensity $\lambda^*|\mathcal{S}|$. To do so, we draw the number of point K from a Poisson distribution $\mathcal{P}(\lambda^*|\mathcal{S}|)$ and the K points locations $\{s_1, \ldots, s_N\}$ uniformly within $\mathcal{S}$. We then treat the $\{s_i\}$ points as input points for our Gaussian process and sample $\{f(s_i)\}$ to obtain $\{\lambda s_i\}$. We finally choose which points to accept or to reject among $\{s_i\}_{i=1}^K$ by thinning i.e. sample a set of uniform variables $\{u_i\}$ and only accept the point s_i when $\lambda(s_i)/\lambda^* > u_i$ for all $i, \ldots, K$.

Algorithm 5 Gaussian Cox process sampling (see Adams et al. [1])

Input: Input space $\mathcal{S}$, GP functions $(\mu(s), K(s, s'))$, upper-bound λ^*, link function $l(\cdot)$
$K \sim Poisson(\lambda^*|\mathcal{S}|)$ ▷ Draw the number of points.
$\{s_i\}_{i=1}^K \sim Uniform(|\mathcal{S}|)$ ▷ Distribute the points uniformly in $|\mathcal{S}|$.
$\{f(s_i)\}_{i=1}^K \sim \mathcal{GP}(\mu_0, \sigma_0)$ ▷ Sample from the GP prior.
$\mathcal{E} \leftarrow \{\emptyset\}$
for $i = 1, 2 \ldots, K$ **do**
    $u_i \sim Uniform(0, 1)$ ▷ Draw uniform variable.
    if $u_i < l(f(s_i))/\lambda^*$ **then** ▷ Apply acceptance rule.
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{s_i\}$ ▷ Add s_i to accepted events.
end for
return $\mathcal{E}$

Appendix E GP-UCB algorithm (Srinivas et al. [5])

Bayesian optimization Assume an optimization problem of the form $f^* = \max_{x \in \mathcal{S}} f(x)$. The Bayesian optimization strategy is to treat the objective function as a random function and place a Gaussian process prior over it. The GP-UCB algorithm then select iteratively the next points in the search space as

$$x_t = \arg\max_{x \in \mathcal{S}} \mu_{t-1}(x) + \beta^{1/2} \sigma_{t-1}(x) \tag{15}$$

where μ_{t-1} and σ_{t-1} are the posterior mean and variance of the Gaussian process conditioned on the previous iterations information $\{(x_1, f(x_1)) \dots, (x_{t-1}, f(x_{t-1}))\}$. The β parameters is a constant used control the degree of exploration. In various scheme β is set to decrease gradually through iterations. Intuitively GP-UCB both explore by choosing a point x with large $\sigma_{t-1}(x)$ and exploits by sampling x with large $\mu_{t-1}(x)$.

Algorithm 6 The GP-UCB algorithm (see Srinivas et al. [5])

Input: Input space $\mathcal{S}$, GP prior (μ_0, σ_0)
for $t = 1, 2 \dots$ **do**
    Choose $x_t = \arg\max_{x \in \mathcal{S}} \mu_{t-1}(x) + \beta^{1/2} \sigma_{t-1}(x)$
    Sample $y_t = f(x_t) + \epsilon_t$
    $\mu_t, \sigma_t \leftarrow \mu_{t-1}, \sigma_{t-1}$ ▷ Perform Bayesian update to the GP mean and variance.
end for

References

[1] MacKay Adams, Murray. Tractable nonparametric bayesian inference in poisson processes with gaussian process intensities, 2005.

[2] Malte J.Rasch Bernhard Scholkopf Alexander Smola Arthur Gretton, Karsten M.Borgwardt. A kernel two-sample test, 2016.

[3] Michael Osborne Stephen Roberts Chris Lloyd, Tom Gunter. Variational inference for gaussian modulated poisson processes, 2015.

[4] Shedler Lewis, P. A. W. Simulation of a nonhomogeneous poisson process by thinning, 1979.

[5] Sham M. Kakade Matthias Seeger Niranjan Srinivas, Andreas Krause. Gaussian process optimization in the bandit setting: No regret and experimental design, 2010.

[6] Seth Flaxman; Michael Chrico; Pau Pereira and Charles Loeffler. Scalable high resolution forecasting of sparse spatio-temporal events with kernel methods: A winning aolution to the nij real-time crime forecasting challenge, 2019.

[7] Jesper Moller Peter McCullagh. The permanental process, 2006.

[8] Carl Edward Rasmussen. Gaussian processes for machine learning, 2005.

[9] L.G Valiant. The complexity of computing the permanent, 2006.

[10] Christian Walder and Adrian Bishop. Fast bayesian intensity estimation for the permanental process, 2018.

[11] Peng Zhu, Li and Xie. Imitation learning of neural spatio-temporal point processes, 2021.

[12] Peng Du Xie Zhu, Li and Song. Learning temporal point processes via reinformcent learning, 2018.