# Efficient Bayesian Nonparametric Modelling of Structured Point Processes

Tom Gunter[*]    Chris Lloyd[*]    Michael A. Osborne    Stephen J. Roberts

Machine Learning Research Group
Department of Engineering Science
University of Oxford
{tgunter,clloyd,mosb,sjrob}@robots.ox.ac.uk

## Abstract

This paper presents a Bayesian generative model for dependent Cox point processes, alongside an efficient inference scheme which scales as if the point processes were modelled independently. We can handle missing data naturally, infer latent structure, and cope with large numbers of observed processes. A further novel contribution enables the model to work effectively in higher dimensional spaces. Using this method, we achieve vastly improved predictive performance on both 2D and 1D real data, validating our structured approach.

## 1   INTRODUCTION

Point processes are effectively used to model a variety of event data, and have also shown a recent popularity within the Machine Learning community as priors over sets. The most fundamental example of such a stochastic model for random sets is the homogenous Poisson process. This is defined via an intensity which describes the expected number of points found in any bounded region of some arbitrary domain. An inhomogenous Poisson process allows the intensity to vary throughout the domain over which the process is defined. As we do not know the functional form of this intensity given only event data, another stochastic process is typically used to model it nonparametrically. This is then termed a doubly-stochastic Poisson process, a type of renewal process also known as a Cox process. In our particular construction, we use transformed Gaussian processes to model the intensity functions of the individual dependent point processes, in such a manner as to enable fully nonparametric Bayesian inference (Adams et al., 2009; Murray et al., 2010). While we only explicitly consider the doubly stochastic Poisson process, any general renewal process (Rao and Teh, 2011) could be incorporated into the framework we define.

There are many occasions when we have multiple point processes which we expect to be dependent: If the domain is temporal, then an example would be individual clients making trades with a specific financial services provider, or individual customers purchasing items from a specific vendor. If the domain is spatial, we might consider different categories of crime defined over some geographic region. Defining a flexible model for inter-process dependency structure, alongside an efficient inference scheme allows us to learn the underlying intensity functions which drive the typical behaviour. These can then be used to make more accurate predictions, especially during periods of unobservability for an individual process.

In order to maximise the flexibility of our approach, we specifically assume that the individual intensity functions arise via a weighted summation of convolutions of latent functions with a kernel. Intuitively this means that we take a small number of latent functions, individually smooth and scale them, and then add them together to yield an intensity function. This approach allows a wide range of intensities to arise from only a few latent functions.

We will present and validate the following novel contributions:

- The first generative model for dependent Cox process data (Section 2).

- An efficient, parallelised inference scheme, which scales benignly with the number of observed point processes (Section 3).

- A new adaptation of thinning (Lewis, 1979), which we term 'adaptive thinning'. This introduces multiple uniformisation levels over the space, making the model viable for higher dimensional spaces and larger datasets (Section 4).

---

[*] Corresponding authors, in alphabetical order.

## 2  THE MODEL

We first formally review the Cox process, before describing the innovative nonparametric Bayesian model outlined in Adams et al. (2009) as the Sigmoidal Gaussian Cox Process (SGCP), which allows a full Gaussian process to be used as a prior over an individual intensity function. We then move on to review the convolution process (Álvarez and Lawrence, 2011), a method of modelling dependent functions and the underlying latent processes which govern them. Our novel combination of these constituent elements represents the first model for dependent Cox point processes.

### 2.1  THE INHOMOGENOUS POISSON PROCESS

For a domain $\mathcal{X} = \mathbb{R}^D$ of arbitrary dimension $D$, we may define an inhomogenous Poisson process via an intensity function $\lambda(x) : \mathcal{X} \to \mathbb{R}^+$, and a Lebesgue measure over the domain, $dx$. The number of events $N(\mathcal{T})$ found over a subregion $\mathcal{T} \subset \mathcal{X}$ will be Poisson distributed with parameter $\lambda_{\mathcal{T}} = \int_{\mathcal{T}} \lambda(x) \, dx$. Furthermore, we define $N(\mathcal{T}_i)$ to be independent random variables, where $\mathcal{T}_i$ are disjoint subsets of $\mathcal{X}$ (Kingman, 1993).

If we bound the region to be considered, and assume there are $K$ observed events, labelled as $\{x_k\}_{k=1}^K$, then the inhomogenous Poisson process likelihood function may be written as

$$p(\{x_k\}_{k=1}^K \mid \lambda(x)) = \exp\left\{-\int_{\mathcal{T}} dx\, \lambda(x)\right\} \prod_{k=1}^K \lambda(x_k). \tag{1}$$

### 2.2  THE SIGMOIDAL GAUSSIAN COX PROCESS

In order to model the intensity nonparametrically, we place a Gaussian Process (Rasmussen and Williams, 2006) prior over a random scalar function $g(x) : \mathcal{X} \to \mathbb{R}$. This means that the prior over any finite set of function values $\{g(x_n)\}_{n=1}^N$ is a multivariate Gaussian distribution, defined by a positive definite covariance function $C(.,.) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and a mean function $m(.) : \mathcal{X} \to \mathbb{R}$. The mean and covariance function are parameterised by a set of hyperparameters, which we denote $\gamma$.

In the SGCP, a Gaussian Process is transformed into a prior over the intensity function by passing it through a sigmoid function and scaling it against a maximum intensity $\lambda^*$: $\lambda(x) = \lambda^* \sigma(g(x))$, where $\sigma(.)$ is the logistic function. This forms the basis of a generative prior, whereby exact Poisson data can be generated from $\lambda(x)$ via thinning (Lewis, 1979), which involves

adding $M$ events, such that the joint point process over the $M + K$ events is homogenous with fixed rate $\lambda^*$.

As we are using an infinite dimensional proxy for $\lambda(x)$, the integral in Equation 1 is intractable. Furthermore, using Bayes' theorem with this likelihood yields a posterior with intractable integrals in both the numerator and denominator. These challenges are overcome by making use of the generative prior, and augmenting the variable set to include the number of thinned points, $M$, and their locations, $\{\tilde{x}_m\}_{m=1}^M$. This then means that the value of the intensity function need only be inferred at the $M + K$ point locations, $\mathbf{g_{M+K}} = \{g(x_k)\}_{k=1}^K \cup \{g(\tilde{x}_m)\}_{m=1}^M$. Noting that $\sigma(-z) = 1 - \sigma(z)$, the joint likelihood over the data, function values and latent variables is

$$\begin{aligned} p(\{x_k\}_{k=1}^K, M, \{\tilde{x}_m\}_{m=1}^M, \mathbf{g_{M+K}} \mid \lambda^*, \mathcal{T}, \theta) = \\ (\lambda^*)^{M+K} \exp\{-\lambda^* \mu(\mathcal{T})\} \\ \times \prod_{k=1}^K \sigma(g(x_k)) \prod_{m=1}^M \sigma(-g(\tilde{x}_m)) \\ \times \mathcal{GP}(\mathbf{g_{M+K}} \mid \{x_k\}_{k=1}^K, \{\tilde{x}_m\}_{m=1}^M, \gamma), \quad (2) \end{aligned}$$

where we have defined $\mu(\mathcal{T}) = \int_{\mathcal{T}} dx$.

Notably, this likelihood equation does not involve any intractable integrals. This means that inference is now possible in this model, albeit subject to the cost of an augmented variable set.

### 2.3  THE CONVOLUTION PROCESS

The convolution process framework is an elegant way of constructing dependent output processes. Instead of assuming the typical instantaneous (Teh et al., 2005) mixing of a set of independent processes to construct correlated output processes, we generalise to allow a blurring of the latent functions achieved via convolution with a kernel, $G(x, z)$, prior to mixing. $z$ is typically defined on the same domain as $x$. If we place a Gaussian process prior over the latent function, the output function turns out to also be a Gaussian process (Álvarez and Lawrence, 2011). Specifically, given $D$ dependant intensity functions $g_d(x)$ and $Q$ latent processes $u_q(x)$, (where typically $Q < D$), the stochastic component of the $d$th intensity is

$$g_d(x) = \sum_{q=1}^Q \int_{\mathcal{T}} G_d(x, z) u_q(z) \, dz. \tag{3}$$

Given full knowledge of the latent functions, the $g_d(x)$ are independent and deterministic. The $G_d(x, z)$ encode the observed process specific characteristics, and the $u_q(z)$ can be thought of as encoding the latent driving forces.

The convolution process has strong links with the Bayesian kernel method, as described in (Pillai et al., 2007). This allows a function $f(x)$ on $\mathcal{X}$ to arise as

$$f(x) = \int_{\mathcal{X}} K(x,z)U(\mathrm{d}z), \qquad (4)$$

where $U(\mathrm{d}z) \in \mathcal{M}(\mathcal{X})$ is a signed measure on $\mathcal{X}$. The integral operator $\mathcal{L}_K : U(\mathrm{d}z) \to f(x)$ maps the space of signed measures $\mathcal{M}(\mathcal{X})$ into $\mathcal{H}_K$, a reproducing kernel Hilbert space (RKHS) defined by the kernel, $K(x,z)$. This mapping is dense in $\mathcal{H}_K$. If we place a Gaussian process prior on the random signed measure $U(\mathrm{d}z)$, rather than directly over $f(x)$, then any draw of $f(x)$ will provably lie in $\mathcal{H}_K$. As $\mathcal{H}_K$ is equivalent to the span of functions expressible as kernel integrals, this approach allows us to properly construct distributions over specific parts of function space. Using prior domain knowledge to restrict inference to plausible areas of function space is valuable, particularly for point process intensities where the likelihood linking function to data is weak and non-trivial, and we wish to only consider smooth intensities while using a sampling based inference scheme.

The convolution process is also known as a latent force model (Alvarez et al., 2009). In this guise, it is used to infer the solution of a differential equation when there is uncertainty in the forcing function. The convolution kernel is the Green's function of a particular differential equation, and the Gaussian process prior is placed on the driving function. This representation lets us consider the latent functions as driving forces, which are viewed through the intensity function specific convolution kernel. The convolution kernel can, for example, be used to model differing speeds of information propagation from the latent factors to each of the observed processes.

It is worth noting that any general Lévy process prior can be used over the latent functions, however in this particular case we use a pure Gaussian process primarily for reasons of tractability.

### 2.4 SPARSE LATENT FUNCTIONS

To make the model tractable, we make use of the property that the intensities are independent conditioned on the latent functions. This is made clear from the perspective of a generative model with only one latent function: we first draw a sample of the object $u(z)$, before solving the integral in equation 3, where uncertainty about $u(z)$ is propagated through the convolution. Now instead of maintaining the full, infinite dimensional object $u(z)$, let us condition on a finite dimensional draw of $u(z)$, $u(Z) = [u(z_1), \ldots, u(z_J)]^T$ where $Z = \{z_j\}_{j=1}^J$. We can then sample from $p(u(z) \mid u(Z))$, as this is a conditional Gaussian distribution,

and use this function to solve the convolution integral. With multiple latent functions we can approximate each $u_q(z)$ by $\mathbb{E}[u_q(z) \mid u_q(Z)]$, replacing Equation 3 with

$$g_d(x) \approx \sum_{q=1}^{Q} \int_{\mathcal{T}} G_d(x,z) \mathbb{E}[u_q(z) \mid u_q(Z)] \, \mathrm{d}z. \quad (5)$$

This is reasonable as long as each $u_q(z)$ is smooth, in the sense that it is well approximated given the covariance function and the finite dimensional sample $u_q(Z)$. In Section 3, we use the approximation in Equation 5 along with the conditional independence assumption to build a tractable inference scheme.

### 2.5 CONSTRUCTING THE MODEL

Let the $Q$ latent functions $u_q(z)$ be modelled as Gaussian processes with Gaussian covariance functions such that

$$u_q \mid \phi_q \sim \mathcal{GP}(0, K_q(z,z')), \qquad (6)$$

where $K_q(z,z')$ is simply the Gaussian kernel

$$K_q(z,z') = \mathcal{N}(z; z', \phi_q). \qquad (7)$$

We use a scaled Gaussian convolution kernel

$$G_d(x,z) = \kappa_d \mathcal{N}(x; z, \theta_d). \qquad (8)$$

This restricts $g_d(x)$ to be at least as smooth as the random draws from $u_q(z)$. The covariance linking $u_q(z)$ to $g_d(x)$ is

$$
\begin{aligned}
K_{f_d, u_q}(x,z) &= \int_{\mathcal{X}} G_d(x,z) K_q(z,z') \mathrm{d}z \\
&= \kappa_d \mathcal{N}(x; z, \theta_d + \phi_q), \qquad (9)
\end{aligned}
$$

and the overall covariance between output functions is

$$
\begin{aligned}
K_{f_d, f_{d'}}(x,x') &= \sum_{q=1}^{Q} \int_{\mathcal{X}} G_d(x,z) \\
&\quad \int_{\mathcal{X}} G_{d'}(x',z') K_q(z,z') \, \mathrm{d}z' \, \mathrm{d}z \\
K_{f_d, f_{d'}}(x,x') &= \sum_{q=1}^{Q} \kappa_d \kappa_{d'} \mathcal{N}(x; z, \theta_d + \theta'_d + \phi_q). \quad (10)
\end{aligned}
$$

We could use this joint covariance function to construct one large joint Gaussian process over all the intensity functions. In doing this, however, the $u_q(z)$ have been implicitly integrated out, and the resulting inference problem will scale computationally as $\mathcal{O}(D^3 N^3)$ with storage requirements of $\mathcal{O}(D^2 N^2)$, where $N = M + K$ is the joint number of events. This

is intractable for any real problem, where we would hope to leverage many dependent point processes to learn a few latent factors with minimal uncertainty.

Let us now define some additional notation: $\mathbf{K}$ denotes a covariance matrix obtained by evaluating the appropriate covariance function at all eligible pairs of data points. Subscripts determine which covariance is used and hence which inputs are valid, e.g. $\mathbf{K}_{g_d,u_q}$ denotes the cross covariance between the $d$th output and $q$th input function. $\mathbf{K}_{g_d,u}$ means stack the $Q$ $\mathbf{K}_{g_d,d_q}$ matrices vertically, $\mathbf{K}_{u,u}$ is a block diagonal matrix where each block corresponds to $\mathbf{K}_{u_q,u_q}$, and $\mathbf{u}$ is the result of stacking the draws from the finite dimensional Gaussians $p(u_q(Z))$ vertically.

We also define: $\phi = \{\phi_q\}_{q=1}^{Q}$, $\kappa = \{\kappa_d\}_{d=1}^{D}$, $\theta = \{\theta_d\}_{d=1}^{D}$, $X_d = \{x_{d,k}\}_{k=1}^{K_d} \cup \{\tilde{x}_{d,m}\}_{m=1}^{M_d}$. If we wish to allow the latent functions to be sampled at different points, then we define a separate $Z_q$ for each $u_q(z)$: $Z_q = \{z_{q,j}\}_{j=1}^{J}$. The set of inputs over all latent functions is then $Z = \{Z_q\}_{q=1}^{Q}$, and similarly for the intensities: $X = \{X_d\}_{d=1}^{D}$.

Notation in place, we determine that given the approximation in Equation 5, the conditional likelihood for $g_d(x)$ is

$$p(g_d \mid u, Z, X_d, \kappa_d, \theta_d, \phi) =$$
$$\mathcal{N}(\mathbf{K}_{g_d,u}\mathbf{K}_{u,u}^{-1}\mathbf{u}, \; \mathbf{K}_{g_d,g_d} - \mathbf{K}_{g_d,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{g_d,u}^{T}). \quad (11)$$

Still conditioning on the latent functions, the joint likelihood over all $D$ intensity functions is then simply

$$p(g_1, \ldots, g_D \mid u, Z, X, \kappa, \theta, \phi) =$$
$$\prod_{d=1}^{D} p(g_d \mid u, Z, X_d, \kappa_d, \theta_d, \phi). \quad (12)$$

Bayes' rule for Gaussians gives us the posterior over the $u_q(Z)$ as

$$p(u_1, \ldots, u_Q \mid g_1, \ldots, g_D, Z, X, \kappa, \phi, \theta)$$
$$= \mathcal{N}(u_1, \ldots, u_Q; \mu_p, \mathbf{\Sigma}_p). \quad (13)$$

Where the mean and covariance are

$$\mathbf{\Sigma}_p = \left[ \mathbf{K}_{u,u}^{-1} + (\mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1})^{T}\mathbf{D}^{-1}(\mathbf{K}_{f,u}\mathbf{K}_{u,u}^{-1}) \right]^{-1}$$
$$\mu_p = \mathbf{\Sigma}_p(\mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1})^{T}\mathbf{D}^{-1}\mathbf{g} \quad (14)$$

and where $\mathbf{D} = \mathbf{K}_{g,g} - \mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{g,u}^{T}$.

The exact form of $\mathbf{D}$ depends on the degree to which we are willing make independence assumptions in order to approximate the Gaussian processes used to model the functions. Naturally the higher the degree of approximation, the more scalable the resulting inference scheme.

Under full dependence, a single event is linked to both inter and intra-process data. We will be assuming that the dependency structure across the $g_d(x)$ is entirely contained by the latent processes, $u_q(z)$. Intuitively, this means that we maintain the full Gaussian process structure for each individual intensity function, while summarising the latent functions via a set of inducing inputs $Z = \{z_j\}_{j=1}^{J}$. This approximation scheme results in a functional form which is similar to what Quiñonero Candela and Rasmussen (2005) call the Partial Independence (PITC) scheme. Importantly it allows inference to scale computationally as $\mathcal{O}(DN^3)$, with storage requirements of $\mathcal{O}(DN^2)$ even in the worst case scenario of $J = N$. Further approximations may be made, and these are especially useful if the number of events per process is large, however for our purposes they are not necessary. For more information on approximation methods for Gaussian processes see Quiñonero Candela and Rasmussen (2005) and Snelson and Ghahramani (2005).

Under the PITC low rank covariance, the resulting form for $\mathbf{D}$ is: $[\mathbf{K}_{g,g} - \mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{g,u}^{T}] \circ \mathbf{M}$, where $\mathbf{M} = \mathbf{I}_N \otimes \mathbf{1}_N$ and $\mathbf{1}_N$ is a $N \times N$ matrix of ones. This may be more familiar as `blkdiag[`$\mathbf{D}$`]`.

## 3   INFERENCE

For each of the $D$ point processes we need to learn $|X_d|$, $X_d$, $\kappa_d$, $\theta_d$, $\lambda_d^*$ and $g_d(x)$. For each of the $Q$ latent functions $u_q(Z)$ and $\phi_q$ must be inferred. $Z_q$ are fixed to an evenly spaced grid which is identical across the latent processes. To find posteriors over all these variables, we choose a Markov Chain Monte Carlo (MCMC) algorithm, as detailed below.

Using the PITC approximation scheme, the likelihood over the point processes factorises conditioned on the latent functions. This means that given $D$ compute units the updates associated with each point process may be made in parallel. This is important as the inference algorithm is computationally bottlenecked by operations associated with learning the locations of the thinning points, $X$.

We now give a recap of the inference scheme from the SGCP for a single point process, while listing our minor modifications. Updates for the latent functions are then given, conditioning on the $D$ intensity functions.

### 3.1   LEARNING THE INTENSITY FUNCTION

Recalling Equation 2, three kinds of Markov transitions are used to draw from this joint distribution: 1) Sampling the number of thinned points, $M$. 2) Sampling the locations of the thinned events, $\{\tilde{x}_m\}_{m=1}^{M}$. 3) Resampling the intensity function, $\mathbf{g}_{M+K}$.

Metropolis-Hastings is used to sample $M$. The probability of insertion/deletion is parameterised by a Bernoulli proposal function: $b(K, M) : \mathbb{N} \times \mathbb{N} \to (0, 1)$, where the parameter has been arbitrarily set to $\frac{1}{2}$. If an insertion is required, a new $x_{M+1}$ is drawn uniformly and at random from $\mu(\mathcal{T})$, and $g(x_{M+1})$ is drawn from the Gaussian process conditioned on the current state. A deletion results in a thinned event $\tilde{x}_m$ being removed at random from $\{\tilde{x}_m\}_{m=1}^M$. The overall transition kernels $q$, and Metropolis-Hastings acceptance ratios, $a$, are:

$$q_{ins}(M + 1 \leftarrow M) = \quad (15)$$
$$\frac{b(K,M)}{\mu(\mathcal{T})} \mathcal{GP}(g(\tilde{x}_{M+1}) \mid \{\tilde{x}_m\}_{m=1}^M, \mathbf{g_{M+K}}),$$

$$a_{ins} = \frac{(1 - b(K, M + 1))\mu(\mathcal{T})\lambda^*}{(M + 1)b(K, M)(1 + \exp(g(\tilde{x}_{M+1})))}, \quad (16)$$

$$q_{del}(M - 1 \leftarrow M) = \frac{1 - b(K, M)}{M} \quad (17)$$

$$a_{del} = \frac{Mb(K, M - 1)(1 + \exp(g(\tilde{x}_m)))}{(1 - b(K, M))\mu(\mathcal{T})\lambda^*}. \quad (18)$$

Sampling the locations of the thinned events also makes use of the Metropolis criterion. For each event $\tilde{x}_m$ a move to $\hat{x}_m$ is proposed via a Gaussian proposal density. A function value $g(\hat{x}_m)$ is then drawn conditioned on the state with $g(\tilde{x}_m)$ removed, denoted $\mathbf{g_{M_-+K}}$. This gives the move acceptance ratio

$$a_{move} = \frac{q_{move}(\tilde{x}_m \leftarrow \hat{x}_m)(1 + \exp(g(\tilde{x}_m)))}{q_{move}(\hat{x}_m \leftarrow \tilde{x}_m)(1 + \exp(g(\hat{x}_m)))}. \quad (19)$$

where $q_{move}$ is the proposal distribution. We use a symmetric Gaussian proposal

$$q_{move}(\hat{x}_m \leftarrow \tilde{x}_m) = \mathcal{N}\left(0, \frac{\mu(\mathcal{T})}{100}\right). \quad (20)$$

To sample the function we opt to use Elliptical Slice Sampling (Murray et al., 2010). This is an algorithm specifically designed for sampling from high dimensional, highly correlated, Gaussian process posteriors. The log conditional posterior over function values is

$$\ln p(\mathbf{g_{M+K}} \mid M, \{x_k\}_{k=1}^K, \{\tilde{x}_m\}_{m=1}^M, \gamma) =$$
$$-\tfrac{1}{2}\mathbf{g_{M+K}}\boldsymbol{\Sigma}^{-1}\mathbf{g_{M+K}} - \sum_{k=1}^K \ln(1 + \exp(-g(x_k)))$$
$$- \sum_{m=1}^M \ln(1 + \exp(g(\tilde{x}_m))) + const. \quad (21)$$

In our case, $\boldsymbol{\Sigma}$ is equal to the covariance in Equation 11, and naturally for each iteration we perform all the above updates in parallel for each observed point process, conditioned on the latent functions.

To infer the posteriors over the Gaussian process hyperparameters, we use Hamiltonian Monte Carlo

---

**Algorithm 1** MCMC Scheme
**Input:** $\{X_k\}_{k=1}^K$, priors.
**repeat**
   **ParFor** $d = 1$ **to** $D$
      Sample thinned events: Equations $16 \to 27$
      Sample locations: Equations $20 \to 19$
      Sample function: Equation 21
      Sample hyperparameters: Equation 21
      Sample $\lambda^*$: Equation 22
   **EndParFor**
   Sample latent functions: Equation 13
   Sample latent hyperparameter: Equation 23
**until** *convergence* is *true*

---

(HMC) (Duane et al., 1987; Neal, 2010), with log-normal priors over each hyperparameter. By placing a Gamma prior with shape $\alpha$ and inverse scale $\beta$ over $\lambda^*$, we infer the posterior conditioned on the thinned and true points using a Gibbs update as follows:

$$\alpha_{post} = \alpha + K + M, \quad \beta_{post} = \beta + \mu\mathcal{T}. \quad (22)$$

### 3.2 LEARNING THE LATENT FUNCTIONS

Conditioning on the point process intensity functions, $g_d(x)$, the latent functions are dependent, with conditional posterior distribution given by Equation 13.

Having drawn new values for each of the $u_q(Z_q)$, we can update the $\phi_q$ using a metropolis-hastings step under the following log conditional posterior which is

$$\ln p(\phi_q \mid u_q(Z_q), Z_q) = -\tfrac{1}{2}u_q(Z_q)\mathbf{K}_{u_q,u_q}^{-1}\tfrac{1}{2}u_q(Z_q)$$
$$-\tfrac{1}{2}\log\det(\mathbf{K}_{u_q,u_q}) + const. \quad (23)$$

The overall procedure is summarised in algorithm 1.

## 4 ADAPTIVE THINNING

In higher dimensional spaces, data is typically concentrated into small, high density sub-domains. Under the current methodology, we must thin the entire empty space to a uniform concentration which matches that of the most dense subregion. If we wish to use Gaussian Process intensities this rapidly becomes infeasible, even under the most radical of sparse approximations (Snelson and Ghahramani, 2005).

Our novel solution to this problem is to model the upper bounded intensity over the space using a piece-wise constant function, where each section takes a fractional proportion of the global upper bound, $\lambda^*$. This preserves the tractability of the integrals in the likelihood and posterior, and does not violate any of the properties of the point process, while simultaneously

allowing empty regions to be thinned to a far lower average density.

Consider Figure 1: this shows both the data and the thinned points, where for the left three quarters of the plot the maximum rate does not exceed 50% of $\lambda^*$. Let us assume we allow the maximum rate to take one of two values for each datapoint: $\frac{1}{2}\lambda^*$ and $\lambda^*$. For each new thinned point we sample an intensity function value, before also sampling an upper bound for the rate from the available levels. This upper bound is at least as great as the current function evaluation at that point.

In this manner we hope to infer that for the majority of Figure 1, the rate may be happily upper-bounded by half the global maximum rate, $\lambda^*$, and hence the bulk of the space may be thinned to a significantly lower density. As a result, the computational burden incurred will be significantly reduced, as far fewer expensive points need be incorporated into our GP.

In our particular implementation, we fix *a-priori* a set of $B$ possible maximum rate 'levels':

$$L = \{l_i \in (0,1] | l_i < l_{i+1}, l_B = 1\}_{i=1}^B. \quad (24)$$

We then augment the variable set to include for each thinned point $\tilde{x}_m$ which rate level $r_m \in \{1 \ldots B\}$ it is currently assigned, where we set $r_m$ such that $\sigma(g(\tilde{x}_m)) \leq l_{r_m}$. This causes the probability of seeing a thinned point $\tilde{x}_m$ under the sigmoid GP to become

$$p(\tilde{x}_m | r_m) = \frac{l_{r_m} - \sigma(g(\tilde{x}_m))}{l_{r_m}}, \quad (25)$$

while the probability of a non-thinned point remains unchanged. Using this relationship we modify the Metropolis acceptance criteria which now become

$$a_{ins} = \frac{(1 - b(K, M+1))\mu(\mathcal{T})\lambda^* l_{r_{M+1}} p(\tilde{x}_{M+1} | r_{M+1})}{(M+1)b(K, M)}, \quad (26)$$

$$a_{del} = \frac{M b(K, M-1))}{(1 - b(K, M))\mu(\mathcal{T})\lambda^* l_{r_m} p(\tilde{x}_m | r_m)}, \quad (27)$$

as well as the likelihood function for $p(g(\mathbf{X_{M+K}}))$, Equation 21.

In principle this scheme could slow mixing, since the function is constrained to lie below the maximum level at each point. By ensuring that there is always some slack, $s$, between the function and the rate level assigned we find that mixing is hardly affected. The slack is incorporated by assigning the rate as follows:

$$r_m \leftarrow \begin{cases} \underset{r}{\text{argmax}} \{\sigma(g(\tilde{x}_m)) \leq l_r \times s\}, & \sigma(g(\tilde{x}_m)) \leq s \\ 1, & \text{otherwise.} \end{cases} \quad (28)$$
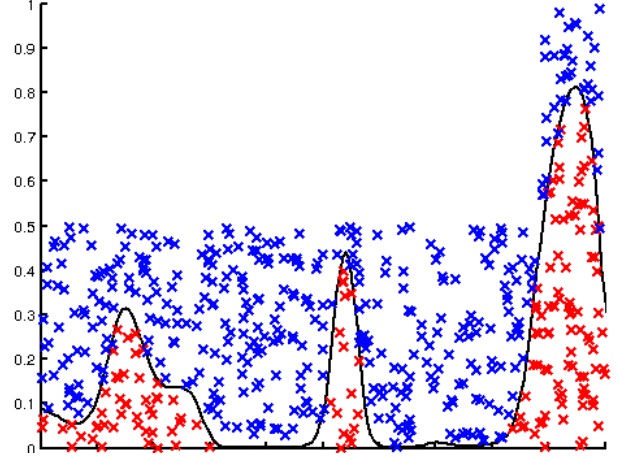


Figure 1: Graphical representation of adaptive thinning: Blue crosses indicate thinned points, red crosses represent data. The black line shows the intensity function. Each point is accepted as data with probability given by $\sigma(g(x_n))$. Fewer thinned points are required in areas of half maximum bound.

We used $s = 0.9$. The rate levels can also change at each iteration during the 'move' step when we compute the new rate level for jittered points and we compose the acceptance criteria as the product of the insertion and deletion criteria $a_{move} = a_{ins} \times a_{del}$.

Finally, when re-sampling $\lambda^*$ we must compute an estimate of the total number of points under a single rate uniformisation of the space. This estimate is readily available because the number of points (including observed data points) with rate $r$ is a Monte-Carlo integral of the proportion of space thinned to rate level $l_r$. Since the number of points in each region is scaled by $l_r$, the estimated total is

$$\hat{N}_{tot} = \sum_{k=1}^K \frac{1}{l_{\bar{r}_k}} + \sum_{m=1}^M \frac{1}{l_{r_m}}, \quad (29)$$

where $\bar{r}_k$ is the notional rate of observed data computed exactly as for the thinned points. The posterior value $\alpha_{post}$ is therefore $\alpha + \hat{N}_{tot}$.

To validate adaptive thinning, we return to the original SGCP and modify it in the manner described above. We perform two experiments: The first in 1D and the second in 2D. In both cases we use 10 known random intensity functions to generate event data: In the 1D case we sample 15 random datasets per function, while in the 2D case we generate 10. One dataset is used to learn the model, the rest are held out for testing purposes. Two metrics of performance are used: L2-norm error as measured against the true intensity function, and average predictive log-likelihood across all held

out test datasets.

In 1D, we run each model for 6 minutes total compute time, in 2D we allow 20 minutes total. In both cases half the time is allocated to burn-in. In 1D the single rate method achieved roughly one sample per second, with the two rate case yielding just under four samples per second, and the four rate approach giving just under 8 samples per second. In 2D the number of points required was larger in all cases, with the multi-rate approach buying a factor of two speedup.

The results, (given in Tables 1 through 4), show that in almost all cases the multi-level approach performs best across both metrics. It is observed that typically the original, homogenous rate approach performs worst of all.

# 5  EMPIRICAL RESULTS

As this is the first model for structured point process data, the approach is initially validated on a synthetic dataset. It is then compared to both the independent SGCP, as well as a state of the art Kernel Density Estimator (Botev et al., 2010) on two real datasets.

## 5.1  SYNTHETIC DATA

Using the convolution process, we sample four intensity functions, using those to sample event data. The variety of intensities which may be observed given a single latent function is notable in Figure 2.

We then average over 2000 iterations after it was determined convergence had been achieved. The resulting learned intensity functions are shown in Figure 3.

It is reassuring that the original latent function is well recovered given only four observed event processes.

## 5.2  REAL DATA

Two datasets were selected to test the model, both of which we considered were likely to exhibit a dependency structure which could be well captured by the convolution process.

- British politicians (MPs) tweet times during the week of Nelson Mandela's death (02/12/13-08/12/13). These were obtained using the Twitter API. Here we considered that there would naturally be a daily periodicity, however, it is not unreasonable to further postulate that some MPs may concentrate their twitter activity into a smaller segment of the day. This behaviour should be well captured by the convolution process.

- NBA player shot profiles for the 2013-2014 season, scraped from the NBA website. Here we se-

lect a diverse subset of four players: Blake Griffin, Damien Lillard, DeMar DeRozen, and Arron Affalo. It was supposed that it might be possible for a single latent function to be blurred to represent a variety of player positions and styles.

### 5.2.1  Twitter Data Results

Four MPs active on twitter were selected at random. Here on we call them MPs A, B, C, and D. We select data from the period covering 02/12/13 through 13/12/13, and randomly partition each dataset into 75% training data, with the remainder being used to evaluate predictive test log-likelihood.

Figure 4 depicts the average learned intensity functions for each MP (red line), along with the one standard deviation bars (grey shading) derived from the function samples. The bottom plot depicts the learned latent driving function in the same manner.

The latent function clearly shows a strong daily period, particularly evident during the working week (02/12/13 was a Monday—corresponding to '1' in Figure 4). Furthermore, the largest two peaks in activity occur on the 3rd and the 5th of December. Potential contributing factors to these two spikes include a public sector strike, and the death of Nelson Mandela respectively.
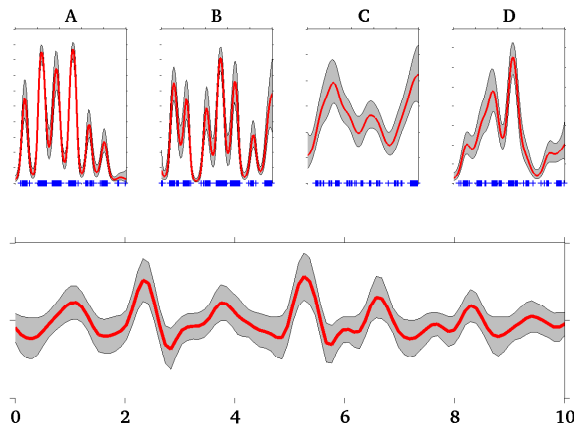


Figure 4: Learned intensities over four MP's tweet data (A, B, C, D); learned latent function at bottom. Actual data shown in blue.

Table 5 gives predictive log-likelihood for the held out data, again evaluated across three approaches: An intensity function learned via Kernel Density Estimation (KDE) (Botev et al., 2010), the SGCP (Adams et al., 2009), and our own structured approach. Both the SGCP and our own approach use two maximum rate levels for adaptive thinning. Each intensity function is modelled using an independent SGCP/KDE. The

Table 1: One dimensional adaptive thinning L2-norm function error. Bold is best.

| Function (1D) | L2 Norm Error | | |
|---|---|---|---|
| | Original | 2 Rates | 4 Rates |
| 1 | 11.6 | 13.0 | **7.1** |
| 2 | 14.6 | 10.5 | **7.2** |
| 3 | 10.6 | 5.0 | **4.9** |
| 4 | 10.5 | **4.7** | 5.1 |
| 5 | 12.4 | **10.1** | 10.4 |
| 6 | 11.1 | 8.2 | **7.6** |
| 7 | **12.0** | 13.7 | 12.5 |
| 8 | 13.0 | **12.0** | 12.8 |
| 9 | 19.6 | **16.4** | 28.8 |
| 10 | 31.4 | **27.2** | 32.6 |

Table 2: One dimensional adaptive thinning average predictive log-likelihood on 14 held out datasets. Bold is best.

| Function (1D) | Predictive Log-Likelihood | | |
|---|---|---|---|
| | Original | 2 Rates | 4 Rates |
| 1 | 373.8 | 381.8 | **388.2** |
| 2 | 626.1 | 644.2 | **650.7** |
| 3 | 274.2 | 285.1 | **288.0** |
| 4 | 435.5 | **457.7** | 456.0 |
| 5 | 877.0 | 885.8 | **889.5** |
| 6 | 995.4 | 1006.6 | **1013.4** |
| 7 | 753.0 | **763.3** | 760.6 |
| 8 | 522.3 | **531.2** | 528.3 |
| 9 | 1840.6 | **1852.9** | 1826.0 |
| 10 | 2328.1 | **2365.8** | 2349.8 |

Table 3: Two dimensional adaptive thinning L2-norm function error. Bold is best.

| Function (2D) | L2 Norm Error | | |
|---|---|---|---|
| | Original | 2 Rates | 4 Rates |
| 1 | 13.3 | 13.4 | **11.3** |
| 2 | 14.3 | **14.3** | 14.7 |
| 3 | 13.5 | 14.7 | **13.5** |
| 4 | 12.5 | 12.9 | **12.0** |
| 5 | 17.9 | **16.6** | 17.8 |
| 6 | **14.4** | 16.2 | 15.3 |
| 7 | 15.1 | **13.7** | 17.5 |
| 8 | 14.6 | **14.4** | 14.8 |
| 9 | 18.3 | 17.1 | **15.6** |
| 10 | 15.4 | **12.9** | 13.6 |

Table 4: Two dimensional adaptive thinning average predictive log-likelihood on 9 held out datasets. Bold is best.

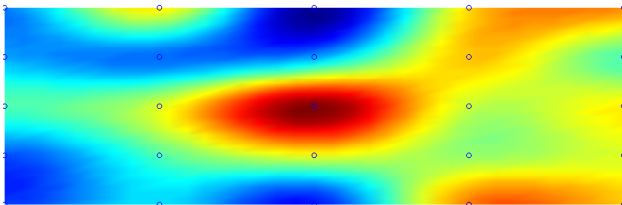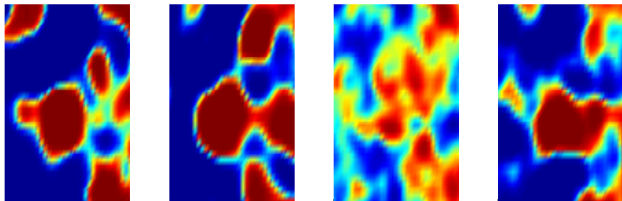| Function (2D) | Predictive Log-Likelihood | | |
|---|---|---|---|
| | Original | 2 Rates | 4 Rates |
| 1 | 2039.6 | 2080.6 | **2203.0** |
| 2 | 2757.9 | 2758.1 | **2829.3** |
| 3 | 2753.2 | 2689.5 | **2827.2** |
| 4 | 2803.2 | 2784.0 | **2933.6** |
| 5 | 2532.4 | **2663.0** | 2572.1 |
| 6 | **3098.2** | 3040.5 | 3054.4 |
| 7 | 3157.5 | **3259.8** | 3075.2 |
| 8 | 2086.9 | **2101.0** | 2087.4 |
| 9 | 5018.1 | 5146.6 | **5185.3** |
| 10 | 2008.1 | **2205.0** | 2174.0 |



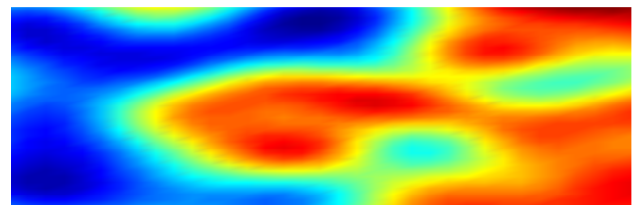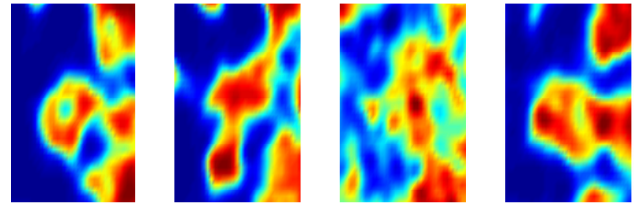Figure 2: Synthetic functions (not showing the sampled events).



Figure 3: Learned functions using 3 rate levels: $\frac{1}{4}\lambda^*$, $\frac{1}{2}\lambda^*$, $\lambda^*$.
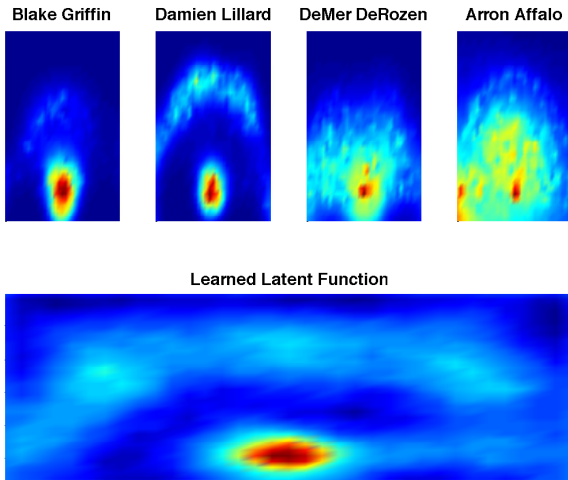
Figure 5: Learned basketball intensity functions using 3 rate levels: $\frac{1}{4}\lambda^*$, $\frac{1}{2}\lambda^*$, and $\lambda^*$.



| Player | Predictive Log-Likelihood | | |
|---|---|---|---|
| | KDE | SGCP | Ours |
| Blake Griffin | -121.8 | 335.6 | **374.7** |
| Damien Lillard | -22.6 | 231.2 | **395.3** |
| DeMer DeRozen | -2.9 | 253.1 | **410.7** |
| Arron Affalo | -260.7 | -76.7 | **84.2** |

Figure 6: Basketball ball shot data (top) and predictive log-likelihood for held out basketball data across models (bottom).

structured approach performs vastly better, suggesting that it is highly appropriate for this type of data.

Table 5: Predictive log-likelihood for held out twitter data across models.

| MP | Predictive Log-Likelihood | | |
|---|---|---|---|
| | KDE | SGCP | Ours |
| A | 177.1 | 176.6 | **469.5** |
| B | -1.3 | 89.6 | **412.9** |
| C | -5.4 | 49.6 | **283.4** |
| D | -67.7 | 38.7 | **293.7** |

### 5.2.2 Basketball Data

For the basketball point shot data, the approach performed particularly well. Each player had around 600 attempted shots, of which we used 400, holding out the rest as test data. We used 3 rate boundaries: $\frac{1}{4}\lambda^*$, $\frac{1}{2}\lambda^*$, $\lambda^*$, and once again averaged over 2000 samples after convergence. We compare predictive log-likelihood to both the SGCP model (using the same set of rate boundaries) and using a rate function estimated via a state of the art KDE by Botev et al. (2010).

Figure 5 depicts the resulting intensity functions. It is clear that the latent function represents a general view of the court hotspots, the hoop and three pointer line are clearly demarcated. Furthermore the intensity functions for each player strongly match what would be expected given their playing style—e.g. Arron Affalo is a 'shooting guard' who is expected to spend the majority of his time inside the three pointer line,

but has a propensity to shoot from the bottom left of the court. These effects are clearly visible on the heat map, less so on the data (see Figure 5.2.1).

As is clearly demonstrated in Table 6, our structured approach to modelling the basketball point data in a fully Bayesian fashion yields a huge improvement over both the independent SGCP as well as a modern kernel density estimator. Another point worth making is that due to the high data density around the hoop for each player, the traditional approach of thinning (used here as well as in the SGCP) would be prohibitively computationally expensive. We are only able to test on this data due to the method of adaptive thinning introduced in this paper.

## 6 CONCLUSION

We have introduced a fully generative model for dependent point processes, alongside an efficient, parallelised inference scheme. We have shown the appropriateness of this model on two real datasets, and introduced a new adaptation of thinning which allows the model to scale to larger datasets and in particular higher dimensional spaces. Future work entails investigating the appropriateness of manually introducing known latent drivers, exploring multiple latent functions, and replacing the MCMC inference scheme with one based on stochastic variational inference (Hensman et al., 2013).

## References

Ryan Prescott Adams, Iain Murray, and David J. C. MacKay. Tractable Nonparametric Bayesian Inference in Poisson Processes with Gaussian Process Intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 9–16, New York, NY, USA, 2009. ACM.

Mauricio A. Álvarez and Neil D. Lawrence. Computationally Efficient Convolved Multiple Output Gaussian Processes. *J. Mach. Learn. Res.*, 12:1459–1500, July 2011. ISSN 1532-4435.

Mauricio A Alvarez, David Luengo, and Neil D Lawrence. Latent Force Models. In *International Conference on Artificial Intelligence and Statistics*, pages 9–16, 2009.

ZI Botev, JF Grotowski, and DP Kroese. Kernel Density Estimation via Diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216 – 222, 1987. ISSN 0370-2693.

James Hensman, Nicoló Fusi, and Neil D. Lawrence. Gaussian Processes for Big Data. *CoRR*, abs/1309.6835, 2013.

J. F. C. Kingman. *Poisson Processes (Oxford Studies in Probability)*. Oxford University Press, January 1993. ISBN 0198536933.

P. A. W. & Shedler G.S. Lewis. Simulation of a Non-homogeneous Poisson Process by Thinning. *Naval Research Logistics Quarterly*, 26:403–413, 1979.

Iain Murray, Ryan P. Adams, and David J. C. MacKay. Elliptical Slice Sampling. *Journal of Machine Learning Research - Proceedings Track*, 9:541–548, 2010.

R. M. Neal. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC Press, 2010.

Natesh S Pillai, Qiang Wu, Feng Liang, Sayan Mukherjee, and Robert L Wolpert. Characterizing the function space for bayesian kernel models. *Journal of Machine Learning Research*, 8(8), 2007.

Joaquin Quiñonero Candela and Carl Edward Rasmussen. A Unifying View of Sparse Approximate Gaussian Process Regression. *J. Mach. Learn. Res.*, 6:1939–1959, December 2005. ISSN 1532-4435.

Vinayak Rao and Yee Whye Teh. Gaussian Process Modulated Renewal Processes. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, Granada, Spain*, pages 2474–2482, 2011.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2nd edition 2006 edition, 2006. ISBN 0-262-18253-X.

E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In *NIPS 18*, 2005.

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric Latent Factor Models. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, volume 10, 2005.