Variational Inference for Gaussian Process Modulated Poisson Processes

Chris Lloyd*  Tom Gunter*  Michael A. Osborne  Stephen J. Roberts
Department of Engineering Science
University of Oxford
{clloyd,tgunter,mosb,sjrob}@robots.ox.ac.uk

Abstract

We present the first fully variational Bayesian inference scheme for continuous Gaussian-process-modulated Poisson processes. Such point processes are used in a variety of domains, including neuroscience, geo-statistics and astronomy, but their use is hindered by the computational cost of existing inference schemes. Our scheme: requires no discretisation of the domain; scales linearly in the number of observed events; and is many orders of magnitude faster than previous sampling based approaches. The resulting algorithm is shown to outperform standard methods on synthetic examples, coal mining disaster data and in the prediction of Malaria incidences in Kenya.

1 INTRODUCTION

Sparse events defined over a continuous domain arise in a variety of real-world applications. The geospatial spread of disease through time, for example, may be viewed as a set of infections which occur in three dimensional space-time. In this work, we will consider data where the intensity (or average incidence rate) of the event generating process is assumed to vary smoothly over the domain. A popular model for such data is the inhomogenous Poisson process with a Gaussian process model for the smoothly-varying intensity function. This flexible approach has been adopted for applications in neuroscience (Sahani et al., 2007), finance (Basu and Dassios, 2002) and forestry (Heikkinen and Arjas, 1999).

However, existing inference schemes for such models scale poorly with the number of data, preventing them from finding greater use. The use of a full Gaussian process in modelling the intensity (Adams et al., 2009) incurs prohibitive $O(N^3)$ computational scaling in the number of data points, $N$. To tackle this problem in practice, many approaches (Rathbun and Cressie, 1994; Mller et al., 1998) discretise the domain, binning counts within each segment. This approach enabled (Cunningham et al., 2008) to achieve $O(N \log N)$ performance. However, the discretisation approach suffers from poor scaling with the dimension of the domain and sensitivity to the choice of discretisation.

We introduce a new model for Gaussian-process-modulated Poisson processes that eliminates the requirement for discretisation, while simultaneously delivering $O(N)$ scaling. We further introduce the first fully variational Bayesian inference scheme for such models, allowing computation many orders of magnitude faster than existing schemes. This approach is shown to provide more accurate prediction than benchmarks on held-out data from datasets including synthetic examples, coal mining disaster data and Malaria incidences in Kenya. The power of our approach suggests many applications: in particular, our fully generative model permits the joint inference of real-valued covariates (such as log-rainfall) and a point process (such as disease outbreaks).

2 INHOMOGENEOUS POISSON PROCESSES

Formally, an inhomogenous Poisson process—also known as a Cox process—is defined via an intensity function $\lambda(x) : \mathcal{X} \to \mathbb{R}^+$. For a domain $\mathcal{X} = \mathbb{R}^R$ of arbitrary dimension $R$, the number of atomic point masses, $N(T)$, found in a subregion $T \subset \mathcal{X}$ is Poisson distributed with parameter $\lambda_T = \int_T \lambda(x) \, dx$, where $dx$ indicates integration with respect to a Lebesgue measure over the domain. In addition for disjoint sub-
sets \( T_i \) of \( \mathcal{X} \), the counts \( N(T_i) \) are independent random variables. This independence is due to the completely independent nature of points in a Poisson process (Kingman, 1993).

If we restrict our consideration to some bounded region, \( \mathcal{T} \), the probability of a set of \( N \) observed points, \( \mathcal{D} = \{x^{(n)} \in \mathcal{T}\}_{n=1}^N \), conditioned on the rate function \( \lambda(x) \) is

\[
p(\mathcal{D} | \lambda) = \exp\left\{ - \int_{\mathcal{T}} \lambda(x) \, dx \right\} \prod_{n=1}^{N} \lambda(x^{(n)}). \tag{1}
\]

We use \( \omega(V) \) to denote the measure of the continuous domain \( V \). In this work we will assume \( \mathcal{T} \) is a box-bounded sub-set of \( \mathbb{R}^R \) with boundaries \( T_r^{\text{min}} \) and \( T_r^{\text{max}} \) in each dimension \( r \) and

\[
\omega(\mathcal{T}) = \int_{\mathcal{T}} 1 \, dx = \prod_{r=1}^{R} (T_r^{\text{max}} - T_r^{\text{min}}). \tag{2}
\]

Using Bayes’ rule to obtain the posterior distribution over the rate function conditioned on the data is a “doubly-intractable” integral

\[
p(\lambda | \mathcal{D}) = \frac{p(\mathcal{D} | \lambda) \exp\left\{ - \int_{\mathcal{T}} \lambda(x) \, dx \right\} \prod_{n=1}^{N} \lambda(x^{(n)})}{\int p(\lambda) \exp\left\{ - \int_{\mathcal{T}} \lambda(x) \, dx \right\} \prod_{n=1}^{N} \lambda(x^{(n)}) \, d\lambda}. \tag{3}
\]

### 2.1 Inferring Intensity Functions

To overcome the challenges posed by the doubly intractable integral (Adams et al., 2009) proposes the Sigmoidal Gaussian Cox Process (SGCP). In the SGCP, a Gaussian process (Rasmussen and Williams, 2006) is used to construct an intensity function prior by passing a random function, \( f \sim \mathcal{GP} \), through a sigmoid transformation and scaling it with a maximum intensity \( \lambda^* \). The intensity function is therefore

\[
\lambda(x) = \lambda^* \sigma(f(x)), \tag{4}
\]

where \( \sigma(\cdot) \) is the logistic sigmoid (squashing) function

\[
\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{5}
\]

To remove the inner intractable integral, the authors augment the variable set to include latent data, such that the joint distribution of the latent and observed data is uniform Poisson over the region \( \mathcal{T} \). While this model works well in practice on small, sparse event data in low dimensions, in reality, it scales poorly with both the dimensionality of the domain and the maximum observed density of points. This is due to: the incorporation of latent, or thinned, data, whose number grows exponentially with the dimensionality of the space; and an \( \mathcal{O}(N^3) \) cost in the number \( N \) of all data (thinned or otherwise).

In (Gunter et al., 2014a), the authors go some way towards improving the scalability of the SGCP, by introducing a further set of latent variables such that the entire space need no longer be thinned uniformly. Instead, they thin to a piecewise uniform Poisson process, maintaining the tractability of the inner integral, and allowing the model to scale to higher dimensional point processes. The authors term this approach “adaptive thinning”.

### 3 MODEL

We construct our prior over the rate function using a Gaussian process. Rather than using a squashing function, we will assume\(^1\) the intensity function is simply defined as \( \lambda(x) = f^2(x) \) where \( f \) is a Gaussian process distributed random function

\[
f(x) \sim \mathcal{GP}(\mu(x), \Sigma(x, x')) \tag{5}
\]

achieving a non-negative prior (Gunter et al., 2014b). Furthermore we will assume that \( f \) is conditionally dependent on another Gaussian process \( u \) evaluated at a set of inducing points \( \mathcal{Z} = \{z^{(m)} \in \mathcal{T}\}_{m=1}^{M} \). We denote the evaluation of \( u \) at these points \( u \), and note \( u \) has distribution \( u \sim \mathcal{N}(\tilde{\mu}, K_{uu}) \).

Using this formulation, the mean and covariance functions of \( f \) are

\[
\mu(x) = k_{zu}K^{-1}_{uu}u, \tag{6}
\]

\[
\Sigma(x, x') = K_{xx'} - k_{zu}K^{-1}_{uu}k_{uz'}, \tag{7}
\]

where \( k_{zu}, K_{xx'}, K_{uu} \) are matrices evaluated at \( x, x' \) and \( \mathcal{Z} \) using an appropriate kernel. We use the exponentiated quadratic (also known as the “squared exponential”) ARD kernel

\[
K(x, x') = \gamma \prod_{r=1}^{R} \exp\left( -\frac{(x_r - x'_r)^2}{2\alpha_r^2} \right). \tag{8}
\]

With this hierarchical formulation the joint distribution over \( \mathcal{D}, f, u \) and \( \Theta \) is

\[
p(\mathcal{D}, f, u, \Theta) = p(\mathcal{D} | f = f^2)p(f | u, \Theta)p(u | \Theta)p(\Theta) \tag{9}
\]

where \( p(\Theta) \) is the (optional) prior on the set of model parameters \( \Theta = \{\gamma, \alpha_1, \ldots, \alpha_R, \tilde{\mu}, \tilde{\alpha}_r\} \). Since the length scales, \( \gamma, \alpha_r \), are constrained to the positive real numbers, log-normal or gamma priors should be used.

For notational convenience we will usually omit conditioning on \( \Theta \).

\(^1\)See Section 5 for a detailed motivation for this choice.
\section{Inference}

We will use variational inference to obtain a bound on the model evidence \( p(\mathcal{D}) \). To achieve this we must integrate out \( f \) and \( u \), but we must also integrate \( f^2 \) over the region \( \mathcal{T} \) due to the integral embedded in the likelihood, Equation 1.

\subsection{Variational Bound}

We begin by integrating out the latent function \( u \), using a variational distribution \( q(u) = \mathcal{N}(u; m, S) \) over the inducing points. Since only \( f \) is conditioned on \( u \) and since \( q(u) \) is conjugate to \( p(f|u) \), we can write down in closed-form the resulting integral:

\[ q(f) = \int p(f|u)q(u)du = \mathcal{G}(f; \bar{\mu}, \bar{\Sigma}), \quad (10) \]

\[ \bar{\mu}(x) = k_{xx}K_{uu}^{-1}m, \]

\[ \bar{\Sigma}(x, x') = k_{xx'} - k_{xx}K_{uu}^{-1}k_{uu'}. \]

We also obtain a KL-divergence term arising from our variational approximation, which is simply the KL-divergence between two Gaussians

\[ \text{KL}(q(u)||p(u)) = \frac{1}{2} \left[ \text{tr} \left( \frac{K_{uu}}{S} \right) - \log \left| \frac{K_{uu}}{S} \right| - M + (\bar{\mu} - \mu)^\top K_{uu}^{-1}(\bar{\mu} - \mu) \right]. \quad (11) \]

Substituting \( q(f) \) into the joint distribution leads to

\[ \log p(\mathcal{D}, f) \geq \log [p(\mathcal{D}|f)p(f)] - \text{KL}(q(u)||p(u)) \quad (12) \]

which we can further lower bound by taking expectations under \( q(f) \). To keep the notation concise we introduce the following identities:

\[ f = f(x) \quad \mu_f = \bar{\mu}(x) \quad \sigma_f^2 = \bar{\Sigma}(x, x) \]

\[ f = f(x^{(n)}) \quad \mu_f = \bar{\mu}(x^{(n)}) \quad \sigma_f^2 = \bar{\Sigma}(x^{(n)}, x^{(n)}). \]

Using the new identities the lower bound on the model evidence is

\[ \log p(\mathcal{D}|\Theta) \geq E_{q(f)} [ \log p(\mathcal{D}|f) ] - \text{KL}(q(u)||p(u)) \]

\[ = E_{q(f)} \left[ - \int_{\mathcal{T}} f^2(x)dx + \sum_{n=1}^{N} \log f^2_n \right] \]

\[ - \text{KL}(q(u)||p(u)) \]

\[ = - \int_{\mathcal{T}} \left[ E_{q(f)}[f^2] + \text{Var}_q(f)[f^2] \right] dx \]

\[ + \sum_{n=1}^{N} E_{q(f)}[\log f^2_n] - \text{KL}(q(u)||p(u)) \]

\[ \triangleq \mathcal{L}. \quad (13) \]

We now have two tasks remaining: we must integrate over the region \( \mathcal{T} \) and compute the expectations \( E_{q(f)}[\log f^2_n] \) at the data points.

\subsection{Integrating Over The Region \( \mathcal{T} \)}

This lower bound has the desirable property that we can take expectations under \( q(f) \) at any specific point, \( x \), of the function value, \( f(x) \), since \( q(f(x)) \) is Gaussian. It is only possible to take useful expectations because: a) we used the conditional GP formulation; b) we have already integrated out the latent function \( u \); and e) we chose a suitable transformation, i.e. \( \lambda(x) = f^2(x) \).

The required statistics for Equation 13 are:

\[ E_q(f)[f^2] = \mu_f = m^\top K_{uu}^{-1}k_{ux}k_{uu}^{-1}m, \quad (14) \]

\[ \text{Var}_q(f)[f^2] = \sigma_f^2 = k_{xx} - \text{Tr}(K_{uu}^{-1}k_{uu}k_{uu}) \]

\[ + \text{Tr}(K_{uu}^{-1}SK_{uu}^{-1}k_{uu}k_{uu}). \quad (15) \]

It is now easy to calculate the integral since only \( k_{ux} = k_{ux}^\top \) is a function of \( x \), leading to the following terms:

\[ \int_{\mathcal{T}} E_q(f)[f^2]dx = \mu_f^\top K_{uu}^{-1}\Psi K_{uu}^{-1}m, \]

\[ \int_{\mathcal{T}} \text{Var}_q(f)[f^2]dx = \gamma(\mathcal{T}) - \text{Tr}(K_{uu}^{-1}\Psi) \]

\[ + \text{Tr}(K_{uu}^{-1}SK_{uu}^{-1}\Psi). \]

For the exponentiated quadratic ARD kernel, the matrix

\[ \Psi = \int K(z, x)K(x, z') \, dz \quad (16) \]

can be calculated by re-arranging the product as a single exponentiated quadratic in \( x \) and \( z \) as follows:

\[ \Psi(z, z') = \int R \exp \left( \frac{-(z_r - z_r')^2}{4\alpha_r} - \frac{(x_r - z_r)^2}{2\alpha_r} \right) dx \cdot \exp \left( \frac{(z_r - z_r')^2}{4\alpha_r} \right) \]

\[ \times \left[ \text{erf} \left( \frac{z_r - z_r'}{\sqrt{\alpha_r}} \right) - \text{erf} \left( \frac{z_r - z_r'}{\sqrt{\alpha_r}} \right) \right] \]

where \( z \equiv [z_1, \ldots, z_R]^\top \) has elements \( z_r = \frac{z_r + z_r'}{2} \).

\subsection{Expectations At The Data Points}

The expectation \( E_{q(f)}[\log f_n^2] \) has an analytical—albeit complicated—solution, expressed as

\[ E_{q(f)}[\log f_n^2] = \int_{-\infty}^{\infty} \log(f^2_n) \, \mathcal{N}(f_n, \mu_n, \sigma_n^2) \, df_n \]

\[ = -G \left( \frac{\mu_n^2}{2\sigma_n^2} \right) + \log \left( \frac{\sigma_n^2}{2} \right) - C, \quad (17) \]

where \( C \approx 0.5772156649 \) is the Euler-Mascheroni constant and \( G \) is defined via the confluent hyper-
geometric function
\[ \sum_{k=0}^{\infty} \frac{(a)_k z^k}{(b)_k k!}, \] (18)
where \((a)_k\) denotes the rising Pochhammer series
\[ (a)_0 = 1, \quad (a)_k = a(a+1)(a+2)\ldots(a+k-1). \]

Specifically \(\tilde{G}\) is a specialised version of the partial derivative of \(1_F_1\) with respect to its first argument:
\[
G(a,b,z) = \frac{\partial}{\partial a} 1_F_1(a,b,z) \quad (19)
\]
\[
\tilde{G}(z) = G \left( 0, \frac{1}{2}, z \right). \quad (20)
\]

\(G\) can be computed using the method of (Ancarani and Gasaneo, 2008), which has a particular solution at \(a = 0\), leading to the following definition of \(G\):
\[
\tilde{G}(z) = 2z \sum_{k=0}^{\infty} \frac{k! z^k}{(2k)(1+z)_k}. \quad (21)
\]
The numerical accuracy of Equation 21 can be improved using an iterative scheme, e.g.
\[
S_0 = A_0 = 1, \quad A_{k+1} = A_k \times \left[ \left( \frac{1 + k}{1 + \frac{z}{2 + k}} \right) \left( \frac{z}{2 + k} \right) \right]
\]
\[
S_{k+1} = S_k + A_{k+1}
\]
and \(\tilde{G}(z) \approx 2zS_K\) for large enough \(K\). In practice however we prefer to use a large multi-resolution look-up table of precomputed values obtained from a numerical-package. As shown in Figure 1, this function decreases very slowly as its argument becomes increasingly negative, so we can easily compute accurate evaluations of \(\tilde{G}(z)\) for any \(z\) by linear interpolation of our lookup table and, as a by-product, we also obtain \(\tilde{G}'\).

We have now defined all the terms in the bound \(L\).

4.4 Optimising The Bound

To perform inference we optimise the variational parameters \(\mathbf{m}, \mathbf{S}\) and the model parameters \(\Theta\) to maximise \(L\). To optimise these simultaneously we construct an augmented vector \(\mathbf{y} = [\Theta^\top, \mathbf{m}^\top, \text{vech}(\mathbf{L})^\top]^\top\) —where vech(\(\mathbf{L}\)) is the vectorisation of the lower triangular elements of \(\mathbf{L}\), such that \(\mathbf{S} = \mathbf{L}\mathbf{L}^\top\). The necessary derivatives are presented in the appendix.

We can compute the maximum-likelihood (ML) solution by optimising \(L(\Theta; \mathbf{y})\) or the maximum-a-posterior (MAP) estimate by maximising \(L(\Theta; \mathbf{y}) + \log p(\Theta)\). (In principle we could also marginalise \(\Theta\) numerically.)

4.5 Locating The Inducing Points

One final part of our model we have so far left unspecified is the number and location of inducing points. In principle, for a given set of parameters \(\Theta\), we will obtain a lower bound for the true GP likelihood for any number of inducing points in any configuration of locations.

In practice, we distribute the inducing points evenly across the domain; we do this for two reasons. The first is because—in contrast to standard GP regression—the accuracy of our solution is not simply governed by the distance between the inducing points and the data points. The variance of \(f(x)\) increases as \(x\) gets further from the inducing points. However the expectation of \(\lambda\) is a function of both the mean and the variance of \(f\). Since we are integrating this variance over the whole domain, we need to reduce the variance at all locations in the domain. This can be achieved with an evenly-spaced grid.

The second reason is that using regularly sampled grids provides computational advantages. When the grid points are evenly spaced, the kernel matrix has Toeplitz structure, and hence allows matrix inversion (and linear solving) in \(O(M \log^2 M)\) time, a fact previously utilised for efficient point processes by Cunningham et al. (2008). Furthermore, when the kernel function is separable across the dimensions (as specified by (8)), the kernel matrix has Kronecker structure which
can further reduce the cost of matrix inversion Osborne et al. (2012). The latter is relevant to all sparse GP applications based on inducing points, however, it is particularly relevant for this application as we are motivated by the doubly intractable nature of Equation 3. In our implementation, we use naïve inversion of the inducing point kernel matrix, $K_{uu}$, resulting in computational complexity of $O(NM^2)$. Hence the impressive computation times reported below can be readily improved with a small amount of additional implementation effort.

5 ALTERNATIVE GP TRANSFORMATIONS

At this point it is worth considering why we have chosen the function transformation $\lambda(x) = f^2(x)$ in preference to other alternatives we might have used. An obvious first choice would be

$$\lambda(x) = \exp(f(x)).$$

(22)

This transformation is undesirable for two reasons. The more obvious of these is that after taking expectations under $q(f)$ we are left with the integral

$$- \int_T \exp \left( \mu_x + \frac{\sigma_x^2}{2} \right) \, dx$$

(23)

which we cannot be computed in closed form. We could approximate the integral using a series expansion, however this would be very difficult with more than a couple of terms and furthermore, since the function is concave, this approximation would not be a lower bound.

The second—and more subtle—reason is that in using this transformation, when we take expectations under $q(f)$ we get

$$E_{q(f)} \{ \exp(f_n) \} = \mu_n.$$  

(24)

Since the mean, $\mu_n$, is not a function of $S$, the variance of the variational distribution $q(u)$, we have effectively decoupled the data from the uncertainty on our variational approximation; this is clearly undesirable.

Another possible candidate is the probit function, $\lambda(x) = \Phi(f(x))$. This can be integrated analytically against the GP prior, however we are again left with a difficult integral over $T$ which is

$$- \int_T \Phi \left( \frac{\mu_x}{\sqrt{1 + \sigma_x^2}} \right) \, dx.$$  

(25)

As the range of this transformation is $[0, 1)$ we would also require additional machinery to infer a scaling variable.

In contrast the square transform presented allows the integral over the region $T$ to be computed in closed form and $E_{q(f)} \{ \log f_n^2 \}$ can be computed very quickly and accurately. Importantly this transformation also maintains the connection between the data and the variational uncertainty.

The square transform is not a one-to-one function; any rate function $\lambda$ may have been generated by $f^2$ or $-f^2$. We can, however, break this non-identifiability using a prior or other means to bias $f$ towards the positive (or indeed negative) solution.

6 EXPERIMENTS

To evaluate our algorithm, we benchmarked against a frequentist kernel smoothing (KS) approach and a fully Bayesian SGCP MCMC sampler. Our test data sets are generative data from the SGCP model and several real-world data sets.

For the synthetic data sets, we first generate a function from a high resolution grid, and then, conditioned on that function, we draw a training dataset and multiple test data sets. We give average and worst case performance for these test data sets. For the real-world data we either: sub-sample without replacement a Poisson-distributed number of points; or allocate each point into a test set with probability 0.5, else it is allocated to the training set.

For our VBPP algorithm, we present two test log-likelihood numbers. The first is simply the lower bound, $\mathcal{L}(\mathcal{H}; y^*)$, evaluated with the learned parameters and the test data set $\mathcal{H} = \{x^{(k)}\}_{k=1}^K$. However, since this bound is not very tight, we find that we do not achieve particularly good predictive likelihoods. To tighten the bound, we therefore adopt the following strategy: having learned the model and variational parameters as before, we take the mean of the variational distribution, i.e. $E[q(u)] = m$, and treat $m$ as a model parameter, that is, we discard its associated uncertainty; this allows us to set $S = 0$. Under this interpretation $m$ is no longer a random variable with a prior and a posterior, and so we can also set $\text{KL}(q(u)\|p(u)) = 0$. We can now re-evaluate the bound which we now denote $\mathcal{L}_0$. In essence the strategy is to use only the mean of the GP posterior to construct our rate function, however we limit this hard decision to the inducing points (therefore $f(x)$, $x \notin \mathcal{Z}$, is still a random function).

6.1 Benchmarks

Our kernel smoothing method is similar to standard kernel density estimation except we use truncated normal kernels to account for our explicit knowledge of the
domain—the latter is referred to as “end-correction” in some literature (Diggle, 1985). The kernel smoother optimises a diagonal covariance, \( \Sigma^* \), by maximising the leave-one-out training objective

\[
\Sigma^* = \arg\max_{\Sigma} \sum_{i=1}^{N} \log \sum_{j \neq i=1}^{N} N_T(x^{(i)}; x^{(j)}; \Sigma).
\] (26)

We can construct the predictive distribution by combining the maximum-likelihood estimates of the size and spatial location of the point process. For the test data set \( \mathcal{H} \) (with \( K! \) permutations) this distribution is

\[
p(\mathcal{H}|D) = K! p(K|D) \prod_{k=1}^{K} p(\tilde{x}^{(k)}|D) \] (27)

where the location density

\[
p(\tilde{x}^{(k)}|D) = \frac{1}{N} \sum_{n=1}^{N} N_T(\tilde{x}^{(k)}; x^{(n)}), \Sigma^*)
\] (28)

is computed using the previously described method and the distribution of the number of points

\[
p(K|D) = \frac{N^K}{K!} \exp(-N)
\] (29)

is simply a Poisson distribution with parameter \( N \). It is straightforward to show that Equation 27 is equivalent to Equation 1 since we can interpret the rate function as

\[
\lambda(x) = \sum_{n=1}^{N} N_T(x; x^{(n)}, \Sigma^*)
\] (30)

and since \( \int_T \lambda(x)dx = N \).

Our SGCP sampler is based on (Adams et al., 2009). Our implementation differs by using elliptical slice sampling to infer the latent functions function \( f \) and we perform hyper-parameter inference using Hybrid Monte-Carlo (HMC). We also use the “adaptive-thinning” method described in (Gunter et al., 2014a) to reduce the number of thinning points required.

### 6.2 Synthetic Data

As an early test, we first draw generative 1D and 2D data from the SGCP, which is a generative model similar to the VBPP. We then attempt to recover the underlying intensity function, using our own approach (VBPP), KDE with end correction (KDE+EC), KDE without end correction (KDE-EC), and the SGCP. Conditioned on \( \sim 80 \) observations, Figures ?? and 2 visualise the inferred intensities: It can be seen in the 1D case that while the difference between the VB(L0) and VB(L) intensities is small, the predictive log-likelihood and RMS error metrics in Tables 1 — 4 are significantly better in the L0 case. In general it is clear that the variational method performs approximately equivalently to the SGCP in the majority of cases, and typically surpasses the results delivered by both KDE methodologies. VBPP arrives at a full posterior over the function in a fraction of the time the SGCP does, albeit typically slightly slower than the maximum-likelihood solution delivered by KDE.

![Figure 2: 2D Synthetic Data. Clockwise from top left: Ground truth, VBPP, KDE+EC, SGCP.](image)
Table 1: Test log-likelihood for 1D synthetic data.

| Function | SGCP Avg. | KDE-EC Avg. | KDE+EC Avg. | VBPP (L) Avg. | VBPP (L0) Avg. |
|----------|-----------|-------------|-------------|---------------|---------------|
| 1        | 209.7     | 148.8       | 202.3       | 147.4         | 207.3         |
| 2        | 68.3      | 41.0        | 61.1        | 32.8          | 65.4          |
| 3        | 4.2       | -4.4        | 0.2         | -7.9          | 3.5           |
| 4        | -3.0      | -8.7        | -7.1        | -13.5         | -3.0          |

Table 2: RMS error and wall-clock time for 1D synthetic data.

| Function | SGCP RMS | KDE-EC RMS | KDE+EC RMS | VBPP (L) RMS | VBPP (L0) RMS |
|----------|----------|------------|------------|--------------|---------------|
| 1        | 2.6      | 124.7      | 3.9        | 0.022        | 3.3           |
| 2        | 1.0      | 120.5      | 3.1        | 0.003        | 2.7           |
| 3        | 1.3      | 112.2      | 1.4        | 0.002        | 1.5           |
| 4        | 0.7      | 108.1      | 1.1        | 0.003        | 0.9           |

Table 3: Test log-likelihood for 2D synthetic data.

| Function | SGCP Avg. | KDE-EC Avg. | KDE+EC Avg. | VBPP (L) Avg. | VBPP (L0) Avg. |
|----------|-----------|-------------|-------------|---------------|---------------|
| 1        | 446.1     | 398.1       | 333.8       | 292.8         | 389.8         |
| 2        | -61.1     | -73.6       | -109.0      | -123.3        | -84.3         |
| 3        | 122.4     | 73.7        | 43.6        | -3.2          | 84.3          |
| 4        | 175.8     | 137.6       | 111.9       | 81.4          | 147.0         |
| 5        | 446.1     | 365.0       | 369.5       | 305.6         | 413.6         |

Table 4: RMS error and wall-clock time for 2D synthetic data.

| Function | SGCP RMS | KDE-EC RMS | KDE+EC RMS | VBPP (L) RMS | VBPP (L0) RMS |
|----------|----------|------------|------------|--------------|---------------|
| 1        | 1.37     | 7547.83    | 2.01       | 0.10         | 1.48          |
| 2        | 0.38     | 1039.65    | 0.66       | 0.01         | 0.46          |
| 3        | 0.88     | 3173.91    | 1.30       | 0.02         | 1.04          |
| 4        | 1.71     | 3773.75    | 1.49       | 0.02         | 1.26          |
| 5        | 2.94     | 6368.44    | 2.29       | 0.05         | 2.02          |
Table 5: Results for Coal Mining Disaster Data

| METHOD  | LOG-LIKE | TIME(s) |
|---------|----------|---------|
| VBPP (L) | -100.3   | 0.7     |
| VBPP (L₀) | -92.2    | 0.7     |
| KS+EC   | -92.6    | 0.0     |
| KS-EC   | -95.9    | 0.0     |
| SCGP    | -96.9    | 417.6   |

6.3.2 Malaria Data

We expect that a major application of the contributions presented in this paper is the joint modelling of disease incidence with correlating factors, in a fully Bayesian, scalable framework. For example, those studying the spread of Malaria often wish to use continuous rainfall measurements to better inform their epidemiological models.

![Malaria incidences in Kenya.](image)

We use examples from the Malaria Atlas Project (map, 2014) to test our scheme. Specifically, we extract 733 incidences of Malaria outbreak documented in Kenya between 1985 and 2010, and run all algorithms excluding the SGCP on half of the resulting dataset, holding out the remainder for testing. Table 6 gives the held out test log-likelihood for the four schemes.

Table 6: Test log-likelihood for 2D Malaria data.

| METHOD  | TEST L-L | RUN TIME (S) |
|---------|----------|--------------|
| KDE-EC  | 199.1    | 224.0        |
| KDE+EC  | 332.1    | 416.6        |

6.3.3 Twitter Data

Finally, we ran the models on the tweet profile of the chairman of the ‘Better Together Campaign’, Alistair Darling, one week either side of the Scottish independence election. Results are shown in Figure 4 and Table 7, where as ever half the data was held out for testing.

Table 7: Test log-likelihood for 1D Twitter data.

| METHOD  | TEST L-L | RUN TIME (S) |
|---------|----------|--------------|
| VBPP (L) | 51.0     | 0.5          |
| VBPP (L₀) | 67.3     | 0.5          |
| KS+EC   | 67.4     | 0.3          |
| KS-EC   | 65.4     | 0.2          |
| SCGP    | 68.1     | 230.0        |

7 FURTHER WORK

Although the performance of the variational Bayesian point process inference algorithm described in this paper improves upon standard methods when used in isolation, it is in its extensions that its utility will be fully realised. Previous work (Gunter et al., 2014a) has shown that hierarchical modelling of point processes—structured point processes—can significantly improve predictive accuracy. In these multi-output models, statistical strength is shared across multiple rate processes via shared latent processes. The method presented here provides a likelihood model for point-process data that can be incorporated as a probabilistic building-block into these larger interconnected models. That is, our fully generative model can readily be extended to additionally incorporate other observation modalities. For example, real-valued observations such as (log-) household income could be modelled along with the intensity function over crime incidents using a variational multi-output GP framework. Future work will be aimed towards developing these variational structured point process algorithms and integration of these techniques to popular Gaussian process tool-kits, such as GPy (the GPy authors, 2014).

8 CONCLUSION

Point process models have hitherto been hindered by their scaling with the number of data. To address this problem, we propose a new model, accommodating non-discretised intensity functions, that permits linear scaling. We additionally contribute a variational Bayesian inference scheme that delivers rapid and accurate prediction. The scheme is validated on real datasets including the canonical coal mining disaster data set and data describing Malaria incidences in Kenya.
References

Malaria Atlas Project. http://www.map.ox.ac.uk/explore/data-modelling/, 2014. [Online; accessed 24-October-2014].

R. P. Adams, I. Murray, and D. J. C. MacKay. Tractable Nonparametric Bayesian Inference in Paussian 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.

L. U. Ancarani and G. Gasaneo. Derivatives of any order of the confluent hypergeometric function _F_1(a,b,z) with respect to the parameter a or b. Journal of Mathematical Physics, 49(6), 2008.

Sankarshan Basu and Angelos Dassios. A cox process with log-normal intensity. Insurance: Mathematics and Economics. 31(2):297-302, Oct 2002. doi: 10.1016/S0167-6687(02)00152-X.

Bradley P. Carlin, Alan E. Gelfand, and Adrian F. M. Smith. Hierarchical Bayesian analysis of changepoint problems. Journal of the Royal Statistical Society. Series C (Applied Statistics), 41(2):138–147, 1992.

John P. Cunningham, Krishna V. Shenoy, and Maneesh Sahani. Fast Gaussian process methods for point process intensity estimation. In Proceedings of the 25th international conference on Machine learning, pages 192–199. ACM, 2008.

P. Diggle. A kernel method for smoothing point process data. Journal of the Royal Statistical Society. Series C (Applied Statistics), 34(2):pp. 138–147, 1985. ISSN 0035-9254.

Paul Fearnhead. Exact and efficient Bayesian inference for multiple changepoint problems. Statistics and Computing, 16(2):203-213, Jun 2006. doi: 10.1007/s11222-006-0450-8.

Tom Gunter, Chris Lloyd, Michael A. Osborne, and Stephen J. Roberts. Efficient Bayesian Nonparametric Modelling of Structured Point Processes. 2014a.

Tom Gunter, Michael A. Osborne, Roman Garnett, Philipp Hennig, and Stephen Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In C. Cortes and N. Lawrence, editors, Advances in Neural Information Processing Systems (NIPS), 2014b.

Juha Heikkinen and Elja Arjas. Modeling a Paussian forest in variable elevations: A nonparametric Bayesian approach. Biometrics, 55(3):738-745, Sep 1999.

R. G. Jarrett. A note on the intervals between coal-mining disasters. Biometrika, 66(1):191-193, 1979. doi: 10.1093/biomet/66.1.191.

J. F. C. Kingman. Paussian Processes (Oxford Studies in Probability). Oxford University Press, jan 1993. ISBN 0198536933.

Jesper Mller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. Log Gaussian Cox Processes. Scandinavian Journal of Statistics, 25(3):451–482, 1998. ISSN 1467-9469.

M. A. Osborne, S. J. Roberts, A. Rogers, and N. R. Jennings. Real-time information processing of environmental sensor network data. ACM Transactions on Sensor Networks, 9(1), 2012. doi: 10.1145/2377979.2378900.

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

Stephen L. Rathbun and Noel Cressie. Asymptotic properties of estimators for the parameters of spatial inhomogeneous Paussian point processes. Advances in Applied Probability, page 122154, 1994.

Maneesh Sahani, M. Yu Byron, John P. Cunningham, and Krishna V. Shenoy. Inferring neural firing rates from spike trains using Gaussian processes, page 329336. 2007.

the GPy authors. GPy: A Gaussian process framework in python. https://github.com/SheffieldML/GPy, 2014.

Appendix

The derivative of the bound with respect to the variational parameters _m_ and _S_ = _LL_\top are:

\[ \frac{\partial L}{\partial m} = -2K_{uu}^{-1}\Psi K_{uu}^{-1}m - K_{uu}^{-1}m \]

\[ \frac{\partial L}{\partial S} = \text{Tr}(K_{uu}^{-1}\Psi K_{uu}^{-1}) - \frac{1}{2} [K_{uu}^{-1} - S^{-1}] \]

\[ \frac{\partial L}{\partial \text{vech}(L)} = \frac{\partial L}{\partial S} \times \frac{\partial S}{\partial \text{vech}(L)} = 2 \text{vech} \left( \frac{dL}{dS} L \right) \]