Correlation Changepoints in Log-Gaussian Cox Processes

CDT Mini Project

Oliver Bartlett

June 11, 2015

Abstract

This report seeks to analyse correlations in Point Processes. It provides an overview of current methods for analysing Point Processes and correlations in Gaussian Process time-series. It reviews Intrinsic and Linear co-regionalisation models, as well as a correlation changepoint model. The use of the Kronecker Product for reducing computational complexity is also explored. All models are tested on the SIGACTS dataset.

1 Introduction

The SIGACTS dataset is a record of ‘Significant Actions’ in the Afghan War spanning from 2006 to 2010. It contains a categorisation of the type of action that takes place such as direct or indirect fire incidents. This report is motivated by the fact that no reliable methods currently exist to analyse the changing correlations between underlying intensities in this dataset.

An area of research which overlaps with this problem is Point Processes, which have applications ranging from basketball analysis [Miller et al., 2014] through to forestry [Stoyan and Penttinen, 2000]. The Log-Gaussian Cox Process model used to infer non-stationary Poisson Processes was introduced by [Møller et al., 1998] and built on the work of [Rathbun and Cressie, 1994]. By making the link to Gaussian Processes [Rasmussen, 2006], another area of active research, many useful features can be included such as changepoints (see [Saatçi et al., 2010] and [Garnett et al., 2010]), although the majority of these methods need adapting to fit within a Poisson Process framework.

Much work has been done to investigate multiple-output Gaussian Processes and the problem of guaranteeing positive semi-definite (PSD) covariance matrices, starting with [Boyle and Frean, 2005] who treat Gaussian Processes as white-noise smoothed with a parameterized kernel in order to handle multiple coupled outputs. [Osborne et al., 2008] use a new method to guarantee PSD covariance matrices and apply it to the area of sensor fusion. [Alvarez and Lawrence, 2009] propose a different approach that models correlated outputs as the convolution between latent Gaussian Proces functions. This work is then strengthened by [Alvarez et al., 2011a] who introduce Variational Bayes for increased model efficiency.

Time-series correlation has also been explored outside the Gaussian Process framework. [Bauwens et al., 2006] gives an overview of Multi-variate Autoregressive conditional heteroscedasticity (MGARCH) models. Wishart Processes [Wilson and Ghahramani, 2010] of-
fer another format for analysing correlated time-series, particularly time-varying covariances between variables.

2 Definitions

2.1 Poisson Process

A Poisson Process is a stochastic process that models events as being generated independently from each other with a given rate. Let $\chi \subseteq \mathbb{R}^d$ be a Euclidean domain and let $\xi \doteq \sum_{i=1}^{n} \delta(x_i)$ be a finite counting measure-valued random variable on $\chi$, where $\delta$ is the Dirac measure, $n \in \mathbb{N}$ is a natural number-valued random variable, and $\{x_i\}_{i=1}^{n} \subseteq \chi$ is a random finite set of points. Let $\lambda : \chi \rightarrow \mathbb{R}^+$ be an arbitrary nonnegative intensity function on $\chi$ and let $\{X_i\}_{i=1}^{n}$ be a finite set of disjoint, bounded Borel-measurable subsets of $\chi$. Define $X \doteq \bigcup_{i} X_i$.

The Poisson Process is characterised by:

$$p(\xi(X) \mid \lambda) = \prod_{i=1}^{m} p(\xi(X_i) \mid \lambda), \quad (2.1)$$

$$p(\xi(X_i) \mid \lambda) = \text{Poisson} \left( \int_{X_i} \lambda \right). \quad (2.2)$$

For the SIGACTS dataset, the input domain $\chi$ is one-dimensional time.

2.2 Cox Process

In a Cox Process, described by [Diggle et al., 1983], the intensity function $\lambda$ is assumed to be generated by a stochastic process which is separate from the Poisson Process. This report assumes that this stochastic process is a Gaussian Process over the log-intensities which guarantees that the intensity stays positive.

2.3 Gaussian Process

A Gaussian Process (GP) is a distribution on the functions $f : \chi \rightarrow \mathbb{R}$ with the property that the distribution of the function values at a finite subset of points $F \subseteq \chi$ is multivariate Gaussian.

A Gaussian Process is completely defined by its first two moments, a mean function $\mu : \chi \rightarrow \mathbb{R}$ and a covariance function $K : \chi \times \chi \rightarrow \mathbb{R}$.

Suppose that there is a GP prior distribution on the function $f : \chi \rightarrow \mathbb{R}$ and a finite set of input points $x$. The prior distribution on $f \doteq f(x)$ is

$$p(f \mid x, \theta) = \mathcal{N}(f; \mu(x, \theta), K(x, x, \theta)) \quad (2.3)$$

where $\mu$ is a vector of mean function values at $x$ and $K$ is a covariance matrix between locations in $x$. The covariance matrix $K$ is constructed from a kernel function $k(x, x')$ where

$$K = \begin{bmatrix} k(x_1, x_1) & \ldots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \ldots & k(x_n, x_n) \end{bmatrix}. \quad (2.4)$$
There are two primary kernels used in this report. The first is the Squared Exponential (SE) kernel which is used to calculate covariance between time signals. It ensures high local correlation and smoothness. The kernel is defined as

$$k_{\text{SE}}(x,x') = \sigma_{\text{SE}}^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)$$  \hspace{1cm} (2.5)$$

where $\sigma_{\text{SE}}$ is the output scale and $l$ is the input length scale.

The second kernel is the noise kernel. It is defined as

$$k_{\text{noise}}(x,x') = \sigma_n^2 \delta(x,x')$$  \hspace{1cm} (2.6)$$

where $\delta$ is the Kronecker Delta and $\sigma_n$ is the magnitude of the noise. It can be used to model measurement uncertainty as well as improving the conditioning of the covariance Matrix. For other kernels see [Duvenaud, 2014] or [Rasmussen, 2006].

### 2.4 Log Gaussian Cox Process

Log Gaussian Cox Processes (LGCPs) were introduced by [Møller et al., 1998]. In a LGCP, a GP prior is placed over the log-intensity $\nu$ in a Cox Process where $\nu \equiv \log(\lambda)$.

$$p(\nu|\theta) \triangleq \mathcal{GP}(\nu; \mu(\cdot; \theta), K(\cdot, \cdot; \theta))$$  \hspace{1cm} (2.7)$$

where $\theta$ contains the hyper-parameters which control the mean and covariance functions. Given observed data $D \triangleq \{x_i\}_{i=1}^n \subset \chi$, the posterior distribution of $\nu$ conditioned on the events is

$$p(\nu | D, \theta) = \frac{\int \int p(D | \nu)p(\nu | \theta)p(\theta) d\theta}{\int \int p(D | \nu)p(\nu | \theta)p(\theta) d\theta d\nu}. \hspace{1cm} (2.8)$$

The Poisson likelihood is given by

$$p(D | \nu) = \exp \left(-\int_\nu \xi \right) \prod_{i=1}^n \exp(\nu(x_i)). \hspace{1cm} (2.9)$$

However, the integral in Equation 2.8 becomes intractable when using this Poisson Likelihood. This can be solved using the Sigmoidal Gaussian Cox Process [Adams et al., 2009] or by discretization and Laplace’s Approximation. This report opts for the latter solution.

To discretise the domain let $\{X_i\}_{i=1}^m$ be a finite partition of $\chi$ where $\cup X_i = \chi$. The log-intensities within the partitions are assumed to be constant and defined as $\{\nu_i\}_{i=1}^m$. These partitions are referred to as ‘bins’ throughout this report.

As a result, Equation 2.9 can be approximated as

$$p(D | \nu) \approx \prod_{i=1}^m \text{Poisson} (\xi(X_i); m(X_i) \exp \nu_i) \hspace{1cm} (2.10)$$

### 2.5 Multiple Output Gaussian Processes

This report makes extensive use of the multiple-output Gaussian Process framework [Osborne et al., 2008] which necessitates updating some of the standard Point Process and Gaussian Process notation defined so far. $l$ is the number of GP outputs or ‘labels’.

The Poisson Process definition (Section 2.1) now contains multiple counting measures $\{\xi_i\}_{i=1}^l$ and associated intensities $\{\lambda_i\}_{i=1}^l$ which act in the same domain $\chi$.  

3
The Gaussian Process Prior (Section 2.3) is now placed on function $f \triangleq f(x, l)$ rather than $f \triangleq f(x)$. It is defined as

$$p(f \mid [x, l], \theta) = \mathcal{N}(f; \mu([x, l], \theta), K([x, l], [x, l], \theta))$$

(2.11)

where the mean is a function of time and the output label. The covariance function is also dependent on $x$ and the output dimension $l$. The exact formulation of this function is considered in Section 3.

The Log-Gaussian Cox Process (Section 2.4) remains largely unchanged, apart for the discretization in Equation 2.10. The constant log-intensities in partitions $\{X_i\}_{i=1}^m$ must be defined for all outputs as $\{\{\nu_i\}_{i=1}^m\}_{j=1}^l$.

### 2.6 Laplace Approximation

Different methods exist to perform inference within the model. [Samo and Roberts, 2014] use Variational Bayes (VB) and inducing points to create a model which scales well with the data. Monte-Carlo based sampling methods can also be used but suffers inefficiency in high dimensions.

This report uses the Laplace Approximation as it is simpler than a VB approach and avoids sampling the posterior. The fact that the posterior is approximated by a Gaussian Process is also a convenient result.

Let $\nu \triangleq [\{\nu_1, ..., \nu_m\}_1, ..., \{\nu_1, ..., \nu_m\}_l]^\top$ represent the constant log-intensities in all partitions $\{X_i\}_{i=1}^m$ and all output dimensions $l$. We approximate the posterior distribution $p(\nu | D, \theta)$ with a Gaussian Distribution obtained by taking a second-order Taylor Expansion of the likelihood around the mode.

Let

$$\hat{\nu} \triangleq \arg\max_{\nu} p(\nu | D, \theta)$$

(2.12)

and

$$\mathcal{H} \triangleq -\nabla\nabla \log p(\nu | D, \theta) \bigg|_{\nu = \hat{\nu}}$$

(2.13)

where $\hat{\nu}$ are the log-intensities which maximise posterior likelihood, and $\mathcal{H}$ is the Hessian of the log-likelihood at $\hat{\nu}$.

The Laplace Approximation states that Equation 2.8 can be approximated as

$$p(\nu | D, \theta) \approx \mathcal{N}(\hat{\mu}, \mathcal{H}^{-1}).$$

(2.14)

The Laplace Approximation can also be used to perform inference on the most likely model hyper-parameters $\hat{\theta}$. The log intensities and model hyper-parameters are concatenated $\mu = \{\nu, \text{vec}(\theta)\}$ and MLE values for hyper-parameters and intensities found simultaneously,

$$\hat{\mu} \triangleq \arg\max_{\mu} p(\mu | D).$$

(2.15)

### 2.7 Gaussian Process Resolution

One of the model parameters which significantly influences the regression is the size of the partitions. Using larger bins removes much of the variance in the data by averaging out the noise. A related effect is that as intensity $\lambda$ increases, the Poisson likelihood $p(D \mid \lambda)$ becomes increasingly Gaussian as a result of the Central Limit Theorem. This improves the suitability of the Laplace Approximation but undermines the motivation for moving away...
from a Gaussian likelihood for which the posterior distribution can be calculated analytically. Figures 2.1 & 2.2 show how the SIGACTS data is regressed differently with different bin sizes. As a result of the averaging out the noise, the length scale $l$ is higher for the monthly regression than the daily regression.

Figure 2.1: 2006 SIGACTS Direct Fire data, binned daily. Inferred hyper-parameters: $l = 42.1$, $\sigma_{SE} = 1.1$.

Figure 2.2: 2006 SIGACTS Direct Fire data, binned monthly. Inferred hyper-parameters: $l = 75.8$, $\sigma_{SE} = 1.87$.

2.8 Kronecker Product

This report also makes significant use of the Kronecker Product where

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \ldots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \ldots & a_{m,n}B \end{bmatrix}$$

(2.16)

$A$ is a $m \times n$ matrix, $B$ can be non-square and $\otimes$ is the Kronecker Product. There are several important results from using the Kronecker Product [Schäcke, 2013].

Firstly that if $A$ and $B$ are both Positive Semi-Definite, $A \otimes B$ will also be PSD. This can be used to construct large PSD covariance matrices from smaller PSD components. This structure can then be exploited when inverting the matrix using the identity:

$$(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$$

(2.17)

The inverse of a Kronecker product is the same as the Kronecker product of the inverses. A similar result holds for Cholesky:
\[
\text{chol}(A \otimes B) = \text{chol}(A) \otimes \text{chol}(B) \quad (2.18)
\]

3 Basic Multiple-Output LGCPs

[Osborne et al., 2008] define a method for calculating fixed correlations in a multi-output GP. A label is introduced to identify which Gaussian Process a value is taken from and a covariance matrix \( K_{\text{label}} \) describes the correlation between these different labels

\[
K_{\text{label}} \triangleq \text{diag}(\lambda) S \text{diag}(\lambda) \quad (3.1)
\]

where \( \lambda \) is a vector of length scales for each variable, and \( S \) is a PSD Matrix. \( S \) can be constructed using Cholesky decomposition \( S = s^T s \) or by spherical decomposition [Pinheiro and Bates, 1996]. For two correlated GPs this can be simplified to

\[
K_{\text{label}} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \quad (3.2)
\]

where \( \rho \) is the correlation between the two processes and the length scales are implemented within the time kernel \( k(x, x') \). \( \rho \) is constrained to lie between -1 and 1 using the mapping \( y = \frac{e^x - 1}{e^x + 1} \).

[Alvarez et al., 2011b] define this method as the Intrinsic Co-regionalisation Model (ICM) where the overall covariance matrix is

\[
K([x, l], [x', l']) = K_{\text{label}} \otimes K_{\text{time}} \quad (3.3)
\]

where \( K_{\text{label}} \) is also known as the co-regionalisation matrix. This method makes use of the Kronecker Product properties in Section 2.8 to speed up the matrix inversion. ICM is used to model the SIGACTS data in Section 3.1.

The Linear Co-regionalisation Model (LCM) is a more flexible version of ICM, which removes the assumption that the co-regionalisation matrix is shared across the separate kernel components. LCM is defined by

\[
k([x, l], [x', l']) = \sum_{q=1}^{Q} B_q k_q(x, x') \quad (3.4)
\]

where \( B_q \) is the coregionalisation Matrix associated with kernel component \( k_q \). LCM is used to model the SIGACTS data in Section 3.2.

3.1 Intrinsic Co-regionalisation Model

To test the model, the SIGACTS dataset was subdivided equally into 500 bins (To keep the regression computationally feasible). The mean function \( \mu(x, \theta) \) was chosen to be 0 as long-term signal forecasting was not deemed a priority.

The covariance matrix \( K(x, x, \theta) \) is calculated by taking the Kronecker product (Equation 3.3) between \( K_{\text{label}} \) (Equation 3.2) and \( K_{\text{time}} \) (Equations 2.4 & 2.5).
Figure 3.1: ICM regression on the SIGACTS data. Top: Correlated LGCP fitted to the Direct Fire component. Middle: Correlated LGCP fitted to the Indirect Fire component. Bottom: Both Posterior LGCPs plotted against each other. Inferred model hyper-parameters: $l = 4.73$, $\rho = 0.97$, $\sigma_{SE} = 40.6$.

Figure 3.1 shows the output of this regression. The inferred correlation $\rho = 0.97$ and input length scale $l = 4.73$. The covariance is not flexible enough to separately capture both lower correlated long term trends and highly correlated local noise. As a result the Laplace-approximated posterior likelihood does not account for the long term trends.

### 3.2 Linear Co-regionalisation Model

LCM overcomes the weaknesses of regressing the SIGACTS data with ICM by assigning different co-regionalisation matrices to the global underlying trend and to the local noise. The mean function and bin sizes were the same as the ICM regression in Section 3.1. The covariance matrix is taken from Equation 3.4 with separately correlated $K_{SE}$ and $K_{noise}$.
components. The full kernel is

\[
K([x, l], [x', l']) = \begin{bmatrix}
1 & \rho_1 \\
\rho_1 & 1
\end{bmatrix} k_{SE}(x, x') + \begin{bmatrix}
1 & \rho_2 \\
\rho_2 & 1
\end{bmatrix} k_{noise}(x, x')
\]  

(3.5)

where \(\rho_1\) and \(\rho_2\) are separate correlations, \(k_{SE}\) and \(k_{noise}\) are defined in Equations 2.5 
& 2.6. One disadvantage of using LCM is that the Kronecker Product properties cannot be used to speed up the matrix inversion as the summation term removes the Kronecker structure. The inferred posterior for this distribution is shown in Figure 3.2. This regression captures underlying trends and doesn’t overfit noise as in the ICM regression.

Figure 3.2: 2006 SIGACTS Direct Fire data regressed with the LCM model. Top: Inferred Direct Fire LGCP. Middle: Inferred Indirect Fire LGCP. Bottom: LGCP Posteriors plotted together. Inferepd hyper-parameters: \(l = 42.3, \sigma_{SE} = 3.04, \rho_1 = 0.996, \rho_2 = 1, \sigma_{noise} = 0.00023.\)
4 Correlation Changepoint LGCP

The disadvantage of Co-regionalisation models is that the correlation between GPs is assumed to be invariant with time. This assumption is often too restrictive for real world applications and datasets such as SIGACTS. This next section attempts to remove this assumption by modelling changing correlations across a changepoint as a first step towards fully dynamic correlations.

[Osborne et al., 2008] use Equation 3.1 to infer correlations between two separate GPs. The same technique can be used to capture correlation changes between two different GPs across a changepoint, by breaking them both into separate GPs before and after the change-point. In total there are 4 Gaussian Processes, labelled $A_1$, $A_2$, $B_1$ & $B_2$ as shown in Figure 4.1. It is assumed that Gaussian Processes A & B use the same partitions.

\[ S = \begin{bmatrix} 1 & c_1 & c_2 & c_4 \\ c_1 & 1 & c_1c_2 + c_1s_1s_2 & c_1c_4 + c_5s_1s_4 \\ c_2 & c_1c_2 + c_1s_1s_2 & 1 & c_3c_4 + s_2s_4(c_3c_5 + c_6s_3s_5) \\ c_4 & c_1c_4 + c_5s_1s_4 & c_3c_4 + s_2s_4(c_3c_5 + c_6s_3s_5) & 1 \end{bmatrix} \] (4.1)

The kernel $K_{SE}(t, t')$ can be used to construct the $K_t$ matrix which defines the prior covariance between each of the times in $A_1$, $A_2$, $B_1$, $B_2$.

\[ K_t = \begin{bmatrix} A_1 & A_2 & B_1 & B_2 \\ K_{11} & K_{12} & K_{11} & K_{12} \\ K_{21} & K_{22} & K_{21} & K_{22} \\ K_{11} & K_{12} & K_{11} & K_{12} \\ K_{21} & K_{22} & K_{21} & K_{22} \end{bmatrix} \] (4.2)

where sub-matrix $K_{ij}$ is the time covariance matrix between datapoints $x_i$ and $x_j$. $\Sigma$ is the overall covariance matrix which combines the time component $K_t$ and label component $K_{label}$:

\[ \Sigma = K_{label} \circ K_t \] (4.3)

where $\circ$ is the Hadamard product.
The posterior distribution of the latent intensity is shown in Figure 4.2. The inferred correlation matrix between the labels is

\[
K_L = \begin{bmatrix}
A_1 & A_2 & B_1 & B_2 \\
1.0000 & -0.5619 & 0.0980 & 3.4094 \\
-0.5619 & 1.0000 & -0.0917 & -1.9156 \\
0.0980 & -0.0917 & 0.0158 & 0.4288 \\
3.4094 & -1.9156 & 0.4288 & 11.6242 \\
\end{bmatrix}
\]  
(4.4)

The inferred hyper-parameters for the SE kernel are: \( l = 61.3 \), \( \sigma_{SE} = 0.51 \). The correlation changepoint \( T_c \) was not a variable parameter of the model, instead it was fixed at 1st July to explore correlations in each half of the year. Although not necessarily visible in Figure 4.2, Equation 4.4 shows that correlation changes across the changepoint are

![Graphs showing regression of SIGACTS Data with correlation changepoint. Top: Posterior LGCP for Direct Fire component. Middle: Posterior LGCP for Indirect Fire component. Bottom: Both Posterior LGCPs plotted against each other. Inferred Model Hyper-parameters: \( l = 61.3 \), \( \sigma_{SE} = 0.51 \), the inferred correlation matrix \( K_L \) is given in Equation 4.4.](image-url)
5 Changepoints with Kronecker Products

Equation 4.3 uses the Hadamard Product to generate the Covariance Matrix \( \Sigma \). The dimensions of \( \Sigma \) are \( mD \times mD \) which is expensive to invert. This section considers whether the Kronecker Product can be used to speed up this inversion as it did in Equation 3.3.

\( K_t' \) is the \( K_t \) matrix (Eq 4.2) with the rows and columns rearranged,

\[
K_t' = \begin{bmatrix}
A_1 & B_1 & A_2 & B_2 \\
K_{11} & K_{11} & K_{12} & K_{12} \\
K_{11} & K_{11} & K_{12} & K_{12} \\
K_{21} & K_{21} & K_{22} & K_{22} \\
K_{21} & K_{21} & K_{22} & K_{22}
\end{bmatrix}. 
\]  

(5.1)

Rather than using the Hadamard product, the Kronecker product can be used to define \( \Sigma' \), which is the same as \( \Sigma \) except with columns and rows rearranged as in Equation 5.1.

\[
\Sigma' = \begin{bmatrix}
1 & K_{L,13} & K_{L,14} \\
K_{L,13} & 1 & K_{L,14} \\
K_{L,12} & K_{L,23} & 1 \\
K_{L,14} & K_{L,43} & K_{L,44}
\end{bmatrix} \otimes K_{11} \quad \begin{bmatrix}
K_{L,12} & K_{L,14} & K_{L,32} & K_{L,34} \\
K_{L,14} & K_{L,24} & K_{L,34} & K_{L,44} \\
K_{L,21} & K_{L,23} & 1 & K_{L,24} \\
K_{L,41} & K_{L,43} & K_{L,44} & 1
\end{bmatrix} \otimes K_{12}
\]  

(5.2)

where \( K_{L,xy} \) is element \((x,y)\) of \( K_{\text{Label}} \) in Equation 3.1. Equations 2.17 and 2.18 cannot be used to speed up the inference for the \( \Sigma' \) matrix because it does not have an overall Kronecker structure. However, using matrix blocking inversion

\[
\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix}
A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\
-(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1}
\end{bmatrix}
\]  

(5.3)

provides some increased efficiency. The block matrix decomposition cannot fully utilise the Kronecker Inversion Lemma in \((D - CA^{-1}B)\) as the subtraction removes the Kronecker structure, this inversion has to be done in full.

However some computation can be saved by rearranging \( \Sigma \) this way, firstly the inversion of \((D - CA^{-1}B)\) only has to be calculated once for the entire matrix. Secondly, the inversion of \( A \) appears in three cells of the Matrix Blocking Formula and could be sped up with the Kronecker Inversion Lemma. This result could be helpful in the case where Gaussian Process \( A \) was calculated over many more bins than Gaussian Process \( B \). For instance in the case where GP \( A \) was a function which needed high resolution results for such as incidents per day, and it exhibited correlation with GP \( B \) which was only known very coarsely, such as climate data.

6 Conclusions and Further Work

This report has put in place methodology for developing covariance matrices which can model correlations within multiple-output Gaussian Processes. The use of Cholesky and Spherical Decompositions as well as the Kronecker Product guarantee that these covariance matrices are positive semi-definite by design. There is a tradeoff between computational efficiency and model flexibility, this tradeoff is most clearly seen when the Kronecker Product structure assumption has to be dropped, for instance in the LCM model (Equation 3.5).

Equation 4.1 shows a PSD correlation matrix which parameterises correlation changes for two GPs across a single changepoint. In order to add dynamic correlations to the model, a kernel which incorporated multiple changepoints could be explored. The locations of these
changepoints could either be fixed in a regular grid, for instance every month, or could be optimised as parameters to reflect the structure of the data. Finding a way to guarantee that the model is not too flexible and doesn’t overfit noise in the data would be a new challenge, one which could perhaps be solved prior distributions on the prior parameters (hyperpriors). It is important to note that adding multiple changepoints would not significantly increase computational costs. The computational bottleneck currently lies in computing the Cholesky decomposition for the Covariance Matrix ($\sim O(ml^3)$, where $m$ is the number of bins and $l$ is the number of output dimensions).Including multiple changepoints would increase the number of parameters in the Laplace argmax optimisation (Equation 2.15) but would not change the dimensions of the covariance matrix which remain $nl \times ml$.

Alternatively other methods of constructing PSD matrices, such as Wishart Processes [Wilson and Ghahramani, 2010] could be explored for Point Processes.

One of the current limitations of the model which is shared with other Point Processes is that it is computationally intensive and doesn’t scale well in large datasets. Several methods exist for doing fast approximate inference in Gaussian Processes, and these could be transferred to Point Process models. For example, [Samo and Roberts, 2014] use Variational Bayes and inducing points to increase the scalability of their single output Point Process regression. These techniques could be extended to the multiple output Point Process domain.

References


[Osborne et al., 2008] Osborne, M. A., Roberts, S. J., Rogers, A., Ramchurn, S. D., and
using computationally efficient multi-output gaussian processes. In *Proceedings of the 7th
international conference on Information processing in sensor networks*, pages 109–120.
IEEE Computer Society.

296.


point process for a longleaf pine forest in southern georgia. *Journal of the American

[Saatç\i et al., 2010] Saatç\i, Y., Turner, R. D., and Rasmussen, C. E. (2010). Gaussian
process change point models. In *Proceedings of the 27th International Conference on
Machine Learning (ICML-10)*, pages 927–934.


