Latent Poisson Process Allocation

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

Abstract

We introduce a probabilistic model for the factorisation of continuous Poisson process rate functions. Our model can be thought of as a topic model for Poisson point processes in which each point is assigned to one of a set of latent rate functions that are shared across multiple outputs. We show that the model brings a means of incorporating structure in point process inference beyond the state-of-the-art. We derive an efficient variational inference scheme for the model based on sparse Gaussian processes that scales linearly in the number of data points. Finally, we demonstrate, using examples from spatial and temporal statistics, how the model can be used for discovering hidden structure with greater precision than standard frequentist approaches.

1 INTRODUCTION

When we observe many real-world phenomena we frequently obtain event or occurrence data. Such data usually consists of a set of points distributed in some spatio-temporal domain. Often we will observe multiple event processes simultaneously (or equivalently a single event process in which each event has an associated label). It is easy to imagine many examples of such situations, for example, the time of purchases of multiple product types or the spatio-temporal distribution of the occurrence of various diseases.

For a variety of purposes, we are typically interested in inferring a function which describes how frequently the events are occurring; in medical or crime prevention applications for example this enables resources to be suitably allocated or anomalous changes in activity to be detected. The Cox process—also known as the doubly-stochastic inhomogeneous Poisson process—is a commonly used model for constructing statistical models of event data, providing the mathematical underpinning for the inference of the associated rate function.

Until recently Bayesian inference of continuous Cox processes required expensive Markov Chain Monte Carlo (MCMC) posterior simulation. It has now been shown that sparse variational Gaussian processes can be used to infer posterior distributions of Cox process rate functions more efficiently without resorting to discretisation of the underlying space (Lloyd et al., 2015a; Matthews et al., 2016).

In this paper we build on this new approach to develop an inference method for multi-output structured point processes, where individual output point processes may share similar characteristics. We name this algorithm Latent Poisson Process Allocation (LPPA), alluding to the similarities between LPPA and Latent Dirichlet Allocation (LDA) (Blei et al., 2003), often used to build topic models of document corpora. In LDA each word in each document is assigned a topic from a set of latent shared topics. In LPPA points are assigned to latent rate functions that are shared across multiple observed point processes. Thus LPPA is conceptually a topic model for Poisson point processes.

LPPA is a continuous analogue of Non-negative Matrix Factorisation (NMF) (Lee and Seung, 2001) and, in particular, bears a resemblance to the fully Bayesian NMF (BNMF) model of Cemgil (2009), since both BNMF and LPPA exploit the infinite divisibility property of the Poisson distribution to apportion data to multiple explanatory factors. However LPPA infers continuous—and not discretised—rate functions and, in addition, LPPA benefits from a smooth spatial prior over the continuous latent factors.

LPPA is also related to the Semi-Parametric Factor Model (SPFM) for multi-task Gaussian process (GP) regression (Teh et al., 2005). Like LPPA, SPFM is is a scaled mixture of latent Gaussian processes except, in

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 41.

the case of the former, the outputs, the (transformed) GP latent functions and mixture parameters are constrained to be positive. Whereas the SPFM uses the Informative Vector Machine (IVM) machinery to achieve computational tractability, LPPA uses a sparse variational approach (Hensman et al., 2013). Furthermore, the LPPA offers efficient inference for point processes, enabling it to be applied to problems for which the SPFM is unsuitable.

LPPA, like the Log-Gaussian Cox Process (LGCP) (Moller et al., 1998) and the Sigmoid-Gaussian Cox Process (SGCP) (Adams et al., 2009), uses transformed Gaussian processes (Rasmussen and Williams, 2006) to construct the prior over the rate function, however LPPA uses the square link function, which results in more tractable integrals.

This work shares a similar motivation to the works of Gunter et al. (2014), Miller et al. (2014) and Lian et al. (2014). Like LPPA, Gunter et al. (2014) uses GPs to model latent rate functions, however the former uses a efficient variational inference scheme rather than the expensive MCMC uniformisation technique employed by the latter. Compared to the approach of Miller et al. (2014), LPPA provides a single integrated model for both rate process smoothing and rate process factorisation.

Lian et al. (2014) is conceptually similar to LPPA. Both methods attempt to factor continuous point processes using a positive weighted sum of latent functions. There are important differences however in the model used to drive the latent processes and the inference methods used, binary semi-Markov Jump Processes (BSMJP) and Forward Filtering and Backward Sampling (FFBS) respectively, which together would seem to limit the approach to 1-dimensional time series. A GP function allows LPPA to extend naturally to higher dimensions and mixed-continuous-and-discrete latent co-ordinate spaces.

LPPA is somewhat related to a body of work on cascading-Poisson processes, otherwise known as Hawkes processes, for example Iwata et al. (2013), Simma and Jordan (2010) and Linderman and Adams (2014). In such processes, events trigger further events and the inference challenge is to infer which events (if any) each event triggers: LPPA as formulated here is not specifically intended for modelling these self-excitatory processes. In addition, LPPA uses a non-parametric model for underlying rate functions, unlike the work on Hawkes processes.

As might be expected due to the generality of the model, LPPA is applicable to a variety of applications, which we explore in Section 4.

1.1 Multivariate Marked Cox Processes

Formally a Cox process—or doubly stochastic inhomogenous Poisson process—over events $\mathscr{X} \triangleq \{\mathbf{x}^{(n)} \in \mathbb{R}^R\}$ is defined via a stochastic intensity function $\lambda(\mathbf{x}) : \mathbb{R}^R \to \mathbb{R}^+$, with an arbitrary domain of dimension R. The number of points, $N(\mathcal{X}_i)$, found in any subregion $\mathcal{X}_i \subset \mathbb{R}^R$ is Poisson distributed with parameter $\lambda_{\mathcal{X}_i} \triangleq \int_{\mathcal{X}_i} \lambda(\mathbf{x})\, \mathrm{d}\mathbf{x}$—where $\mathrm{d}\mathbf{x}$ indicates integration with respect to the Lebesgue measure over the subregion—and for disjoint subsets $\mathcal{X}_i$, $\mathcal{X}_j$, the counts $N(\mathcal{X}_i)$, $N(\mathcal{X}_j)$ are independent. This independence is due to the completely independent nature of points in a Poisson process (Kingman, 1993).

A *marked* Cox process over events $\mathscr{M} \triangleq \left\{\left(\mathbf{x}^{(n)}, A(\mathbf{x}^{(n)})\right) \mid \mathbf{x}^{(n)} \in \mathbb{R}^R,\ A(\mathbf{x}^{(n)}) \in \mathcal{T}\right\}$ extends a Cox process by associating with each point, $\mathbf{x}$, an additional piece of information $A(\mathbf{x}) \in \mathcal{T}$ called a mark. The form of mark itself is very general; it can be a discrete random variable, real valued random variable or indeed another point process. The mark assigned at any point can depend on the location and the value of the rate function at that location.

We consider a multivariate (discrete) mark set $\mathcal{T} \triangleq \{1, \ldots, T\}$. As the marks are discrete there will be a rate process, $\lambda_t(\mathbf{x})$, associated with each mark t. Furthermore, since the set of all points, $\mathscr{M}$, is equal to the union of all sets $\mathscr{M}_t \triangleq \left\{\left(\mathbf{x}^{(n)}, t\right) \mid \mathbf{x}^{(n)} \in \mathbb{R}^R\right\}$, the overall rate process must be the sum of the individual rate processes. Therefore $\lambda(\mathbf{x}) = \sum_{t=1}^{T} \lambda_t(\mathbf{x})$ and, using the Poisson-multinomial connection, the probability of mark t at a point $\mathbf{x}$ is $p(t; \mathbf{x}) = \lambda_t(\mathbf{x})/\lambda(\mathbf{x})$. In general the individual rate functions λ_t may not be independent, although in the model we develop in the next section we will assume they are.

In this framework, the probability density of a set of N observed marked points $\mathscr{M} = \{(\mathbf{x}^{(n)}, A^{(n)})\}_{n=1}^N$, where $A^{(n)} = A(\mathbf{x}^{(n)})$, in some bounded region, $\mathcal{X}$, factorises conditioned on the rate processes[1]

$$p(\mathscr{M} \mid \lambda_{1:T}) = p(\{\mathbf{x}^{(n)}\}_{n=1}^N \mid \lambda_{1:T}) \times p\left(\{A^{(n)}\}_{n=1}^N \mid \{\mathbf{x}^{(n)}\}_{n=1}^N, \lambda_{1:T}\right). \quad (1)$$

The probability density of the points is

$$p(\{\mathbf{x}^{(n)}\}_{n=1}^N \mid \lambda_{1:T}) = \exp\left\{-\int_{\mathcal{X}} \lambda(\mathbf{x})\, \mathrm{d}\mathbf{x}\right\} \prod_{n=1}^{N} \lambda(\mathbf{x}^{(n)}), \quad (2)$$

and, using the Poisson-multinomial connection, the

[1] We use the MATLAB-like notation $p(a_{1:I} \mid b_{1:J})$ to denote $p(a_1, \ldots, a_I \mid b_1, \ldots, b_J)$.

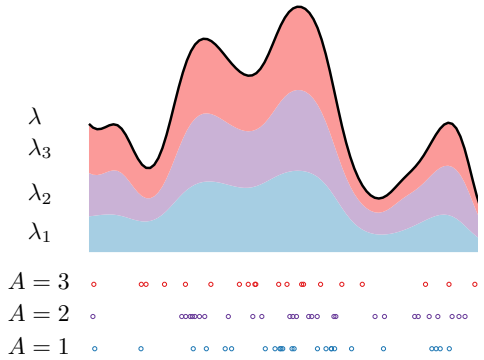

Figure 1: Rate functions are correlated via a convolution process.

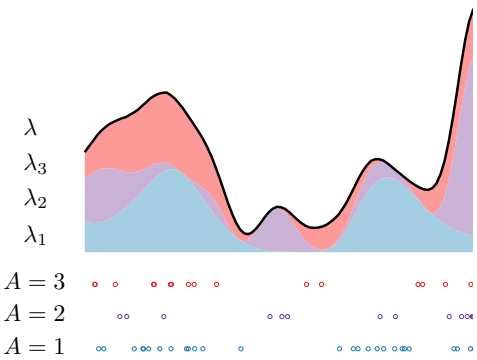

Figure 2: Rate functions are uncorrelated square-transform Gaussian processes.

probability of the N labels is

$$p\big(\{A^{(n)}\}_{n=1}^{N} \mid \{\mathbf{x}^{(n)}\}_{n=1}^{N}, \lambda_{1:T}\big) = \prod_{n=1}^{N} \frac{\lambda_{A^{(n)}}(\mathbf{x}^{(n)})}{\lambda(\mathbf{x}^{(n)})}. \tag{3}$$

A standard inference task for multiple point processes would be to learn the labelling distribution and unobserved point process intensity λ. For independent rate functions, this reduces to simply inferring the rates functions λ_t from subset of points marked with the corresponding mark t.

If the rate functions are not independent, then we may seek to refine our model by constructing a structural prior which allows statistical strength to be shared between the point processes associated with each label. The latter approach was validated by Gunter et al. (2014), using a convolution process to tie the rate processes together, as shown in Figure 1.

In this work we will assume the marks $\mathcal{T}$ are unobserved latent variables to be inferred. Due to a lack of observability, this is impossible for a single point process. To increase observability we need to observe multiple point process in which the latent rate functions are present in linearly independent proportions. We will designate each of these point processes a separate *output* yielding a dataset $\mathscr{D}_s \triangleq \{\mathbf{x}^{(s,n)} \in \mathcal{X}\}_{n=1}^{N_s}$ and each of which will be tied to its own set of output rate functions $\lambda_{s,t}$. Since these outputs are themselves marks in $\mathcal{S} = \{1, \ldots, S\}$, their superposition is a marked point process. There are therefore two distinct sets of marks: observed marks in $\mathcal{S}$ corresponding to the outputs and unobserved marks in $\mathcal{T}$ corresponding to the latent function or topic.

1.2 Permanental Point Processes

There are a variety of options for constructing the strictly non-negative stochastic rate function $\lambda(x)$, which drives the point process of a given mark. One common approach is to transform a Gaussian process through a link-function, where typical choices include the exponential (Kom Samo and Roberts, 2015), and the sigmoid function (Adams et al., 2009). We follow (Lloyd et al., 2015a) in using the square transform.

Constructing a rate function as a sum of square transformed, zero-mean independent Gaussian processes results in a particular sub-class of Cox processes known in the mathematical probability literature as *permanental* point processes (Eisenbaum and Kaspi, 2009; Hough et al., 2006).

The square transform allows efficient variational Bayesian inference machinery that is entirely tractable—this is not the case for the other two transforms. The square transform places prior mass on a larger set of translation invariant Cox processes than the other two link-functions. The reason for this is as follows: as we break the independence assumption inherent in the homogeneous Poisson process and instead introduce an increasing amount of positive correlation between disjoint neighbouring sub-sets of the space, the resulting point process will exhibit ever stronger clustering behaviour for a given configuration of points. In order to model that range of point processes, from nearly homogeneous to strongly clustered, we need a link function that results in a transformed prior which has good dynamic range, but can also achieve both very high and very low function values, without breaking dependence between nearby function values. The squared transform has high dynamic range, and furthermore is unique amongst the set listed in being able to easily model low values—including numerical zero if necessary.

From Figures 1 and 2, we can see that this makes intuitive sense: the increased dynamic range afforded by the square transform leads to a better ability to

model high and low function values, which in turn enables strong local correlations between neighbouring subsets of the space—a characteristic of permanental point process.

2 MODEL

We will assume the intensity of the t^{th} topic of the s^{th} output is $\lambda_{s,t}(\mathbf{x}) = \gamma_{s,t} f_t^2(\mathbf{x})$, where the functions f_t are independent Gaussian process distributed random functions. We constrain the output length scales $\gamma_{s,t}$ to being positive and thus the rate functions are a positive mixture of positively valued latent functions f_t^2.

We condition each latent function f_t at a set of *inducing* points $\mathcal{Z} \triangleq \{\mathbf{z}^{(m)} \in \mathcal{X}\}_{m=1}^{M}$ and we denote the evaluation of f_t at these points $\mathbf{u}_t \sim \mathcal{N}(\vec{1}\bar{u}_t, \mathbf{K}_{zz})$. Therefore $f_t|\mathbf{u}_t \sim \mathcal{GP}\big(\mu_t(\mathbf{x}), \Sigma_t(\mathbf{x}, \mathbf{x}')\big)$ is a Gaussian process with mean function $\mu_t(\mathbf{x}) = \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{u}_t$ and covariance function $\Sigma_t(\mathbf{x}, \mathbf{x}') = \mathbf{K}_{xx'} - \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'}$, where $\mathbf{k}_{xz} = \mathbf{k}_{zx}^\top$, $\mathbf{K}_{xx'}$, $\mathbf{K}_{zz}$ are matrices evaluated at $\mathbf{x}$, $\mathbf{x}'$ and $\mathcal{Z}$ using a suitable kernel function. For notational convenience, we have assumed that all latent functions f_t share the same set of inducing points $\mathcal{Z}$, although it is entirely possible to relax this constraint. In this work we use the exponentiated quadratic (also known as the "squared exponential") Automatic Relevance Determination (ARD) kernel with parameters $\alpha_{1:R}$ (Appendix A.1).

Combining the Cox process likelihood, Equation 2, and mark likelihood, Equation 3, using indicator variables, square transform Gaussian process rate functions and output (square) length scales $\gamma_{s,t}$ gives[2]

$$p(\mathscr{D}_s, A_s \mid f_{1:T}, \Theta) = \prod_t \exp\left(-\int \gamma_{s,t} f_t^2(\mathbf{x}) \mathrm{d}\mathbf{x}\right) \times \prod_n \left[\gamma_{s,t} f_t^2\big(\mathbf{x}^{(s,n)}\big)\right]^{\mathbb{1}\{A_s^{(n)}=t\}}. \quad (4)$$

The joint distribution of $\mathscr{D}_{1:S}$, $f_{1:T}$, $\mathbf{u}_{1:T}$ and $A_{1:S}$ in this hierarchy is

$$p(\mathscr{D}_{1:S}, A_{1:S}, f_{1:T}, \mathbf{u}_{1:T} \mid \Theta) = \prod_t p(f_t \mid \mathbf{u}_t)\, p(\mathbf{u}_t) \times \prod_s p(\mathscr{D}_s, A_s \mid f_{1:T}), \quad (5)$$

where $\Theta \triangleq \{\Gamma, \alpha_{1:R}, \bar{u}_{1:T}\}$ is the set of model parameters and $\Gamma \in \mathbb{R}_+^{S\times T}$ is a matrix of output length scales with elements $\gamma_{s,t}$. For notational convenience we will often omit conditioning on Θ.

[2]We use $\sum_n$ as shorthand for $\sum_{n=1}^{N_s}$ and $\sum_t$ for $\sum_{t=1}^{T}$ and $\sum_s$ for $\sum_{s=1}^{S}$ and analogously for products.

3 VARIATIONAL INFERENCE

We will use variational inference to obtain a bound on the model evidence $p(\mathscr{D}_{1:S})$. To achieve this we take the following steps: firstly integrate out the inducing points $\mathbf{u}_t$ and marginalise the rate functions f_t^2 (Section 3.1) to obtain an uncollapsed lower bound; then integrate this uncollapsed bound over the region $\mathcal{X}$ (Section 3.2); before collapsing the indicator variables A_s (Section 3.3).

3.1 The Uncollapsed Bound

We begin by integrating out the latent function variables $\mathbf{u}_{1:T}$ using a variational distribution $q(\mathbf{u}_{1:T}) = \prod_t q(\mathbf{u}_t)$. In contrast to standard variational inference approaches we bring both $q(\mathbf{u}_{1:T})$ and $p(f_{1:T} \mid \mathbf{u}_{1:T})$ outside of the logarithm giving the uncollapsed bound (also see §A.2):

$$\begin{aligned}\log p(\mathscr{D}_{1:S}, A_{1:S}|\Theta) &\geq \mathbb{E}_{q(f_{1:T})}\big[\log p(\mathscr{D}_{1:S}, A_{1:S} \mid f_{1:T})\big] \\ &\quad - \mathrm{KL}\big(q(\mathbf{u}_{1:T}) \,\big\|\, p(\mathbf{u}_{1:T})\big) \\ &\triangleq \mathcal{L}(\mathscr{D}_{1:S}, A_{1:S}; \Theta). \end{aligned} \quad (6)$$

Since the likelihood is not directly dependent on $\mathbf{u}_{1:T}$ we can integrate it out before taking expectations. As $p(f_t \mid \mathbf{u}_t)$ and $q(\mathbf{u}_t)$ are conjugate, we can write $q(f_t) = \int p(f_t \mid \mathbf{u}_t)\, q(\mathbf{u}_t) \mathrm{d}\mathbf{u}_t = \mathcal{GP}\big(f_t; \tilde{\mu}_t(\mathbf{x}), \tilde{\Sigma}_t(\mathbf{x}, \mathbf{x}')\big)$, (7)

where $\tilde{\mu}_t(\mathbf{x}) = \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{m}_t$ and $\tilde{\Sigma}_t(\mathbf{x}, \mathbf{x}') = \mathbf{K}_{xx'} - \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'} + \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{S}_t\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'}$ and $q(f_{1:T}) = \prod_t q(f_t)$. The last term in Equation 6 is the Kullback-Leibler divergence between T pairs of independent Gaussian distributions (§A.3).

We expand Equation 6 using (4) to give[3]

$$\begin{aligned}&\mathbb{E}_{q(f_{1:T})}\big[\log p(\mathscr{D}_{1:S}, A_{1:S} \mid f_{1:T})\big] = \quad (8)\\ &\sum_s \Bigg[-\sum_t \gamma_{s,t} \int_{\mathcal{X}} \Big(\mathbb{E}_{q(f_t)}[f_{t,x}]^2 + \mathrm{Var}_{q(f_t)}[f_{t,x}]\Big)\mathrm{d}\mathbf{x} \\ &+ \sum_n \sum_t \mathbb{1}\{A_s^{(n)} = t\}\big(\log(\gamma_{s,t}) + \mathbb{E}_{q(f_t)}[\log f_{s,t,n}^2]\big)\Bigg].\end{aligned}$$

The integral $\mathbb{E}_{q(f_t)}[\log f_{s,t,n}^2]$ has an analytic solution

$$\begin{aligned}\mathbb{E}_{q(f_t)}[\log f_{s,t,n}^2] &= -\tilde{G}\left(-\frac{\tilde{\mu}_{s,t,n}^2}{2\tilde{\sigma}_{s,t,n}^2}\right) + \log\left(\frac{\tilde{\sigma}_{s,t,n}^2}{2}\right) - C \\ &= \mathfrak{G}_{s,t,n} \end{aligned} \quad (9)$$

where $C \approx 0.5772156$ is the Euler-Mascheroni constant and $\tilde{G}$ is a specialised version of a partial derivative of the confluent hyper-geometric function (Ancarani and Gasaneo (2008), §A.4).

[3]We use the following shorthand definitions: $f_{t,x} \triangleq f_t(\mathbf{x})$, $\tilde{\mu}_{t,x} \triangleq \tilde{\mu}_t(\mathbf{x})$, $\tilde{\sigma}_{t,x}^2 \triangleq \tilde{\Sigma}_t(\mathbf{x}, \mathbf{x})$, $f_{s,t,n} \triangleq f_t(\mathbf{x}^{(s,n)})$, $\tilde{\mu}_{s,t,n} \triangleq \tilde{\mu}_t(\mathbf{x}^{(s,n)})$, $\tilde{\sigma}_{s,t,n}^2 \triangleq \tilde{\Sigma}_t(\mathbf{x}^{(s,n)}, \mathbf{x}^{(s,n)})$

3.2 Integrating over the region $\mathcal{X}$

Due to the integral embedded in the likelihood, Equation 4, we demand the following integrals over the region $\mathcal{X}$, where $|\mathcal{X}| = \int_{\mathcal{X}} \mathrm{d}\mathbf{x}$:

$$\int_{\mathcal{X}} \mathbb{E}_{q(f_t)}[f_{t,x}]^2 \mathrm{d}\mathbf{x} \;= \mathbf{m}_t^\top \mathbf{K}_{zz}^{-1} \Psi_{zz} \mathbf{K}_{zz}^{-1} \mathbf{m}_t, \tag{10}$$

$$\int_{\mathcal{X}} \mathrm{Var}_{q(f_t)}[f_{t,x}] \mathrm{d}\mathbf{x} = |\mathcal{X}| - \mathrm{Tr}(\mathbf{K}_{zz}^{-1}\Psi_{zz}) + \mathrm{Tr}(\mathbf{K}_{zz}^{-1}\mathbf{S}_t\mathbf{K}_{zz}^{-1}\Psi_{zz}). \tag{11}$$

The matrix Ψ_{zz} is constructed using the function $\Psi(\mathbf{z}, \mathbf{z}') = \int_{\mathcal{X}} K(\mathbf{z}, \mathbf{x}) K(\mathbf{x}, \mathbf{z}') \mathrm{d}\mathbf{x}$. Ψ can be computed in closed form for the ARD kernel used in this work (§A.5), as well as for other kernels including the spectral kernel (Wilson et al., 2014).

3.3 Collapsing the Bound

The bound (6) contains a large number of multivariate indicator variables $A_s^{(n)}$. The standard variational inference approach to this problem would be to marginalise these variables using a variational distribution $q(A_{1:S})$ and to update $q(\mathbf{u}_{1:T})$, $q(A_{1:S})$ and Θ alternately using co-ordinate ascent 'E' and 'M'-steps. Instead we prefer to collapse out the indicator variables and update the variational parameters and model parameters in a single joint optimisation.

Collapsed variational Bayes has a couple of primary benefits: firstly it reduces the number of variables which must be explicitly updated via a marginal gradient step at each iteration; secondly, as we have analytically marginalised (in this case a large subset of) the unknown variables, the implicit updates of those unknown variables will occur with greater efficiency (in the sense that they will converge to a solution faster) (Hensman et al., 2012).

To do this we first note that we can write the bound (6) as the sum of a set of variables $\mathfrak{A}_{s,t,n} = \log(\gamma_{s,t}) + \mathfrak{G}_{s,t,n}$ which multiply the indicator variables $A_s^{(n)}$ and a term $\mathfrak{B}$, that does not, resulting in the compact definition:

$$\mathcal{L}(\mathscr{D}_{1:S}, A_{1:S}; \Theta) = \mathfrak{B} + \sum_{s,t,n} \mathbb{1}\{A_s^{(n)} = t\} \mathfrak{A}_{s,t,n}. \tag{12}$$

To collapse the bound we sum over all the possible assignments to each of the allocation variables[4]:

$$\begin{aligned} \log p(\mathscr{D}_{1:S}|\Theta) &= \log \sum\nolimits_{A_{1:S}} p(\mathscr{D}_{1:S}, A_{1:S}|\Theta) \\ &\geq \mathfrak{B} + \sum\nolimits_s \sum\nolimits_n \log \sum\nolimits_t \exp \mathfrak{A}_{s,t,n} \\ &\triangleq \mathcal{L}(\mathscr{D}_{1:S}; \Theta) \end{aligned} \tag{13}$$

[4] A more complete derivation is given in §A.6.

3.4 Kronecker Structure

Since our GP kernel, is—by necessity—separable across input dimensions, e.g. Equation 18, we can construct our kernel matrices to have Kronecker structure. Such an approach was previously used with Poisson-likelihood GP models by Flaxman et al. (2015), albeit outside of a variational framework. To achieve this structuring we begin by introducing a separate set of inducing points $\mathcal{Z}_r \triangleq \left\{ z_r^{(m)} \in [\mathcal{X}_r^{\mathrm{Min}}, \mathcal{X}_r^{\mathrm{Max}}] \right\}_{m=1}^{M_r}$ for each dimension r, so that the overall inducing point set is the cross product of these sets, i.e. $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_R$ and $M = \prod_r M_r$.

Using these inducing points we can construct the matrices $\mathbf{K}_{zz}$, Ψ_{zz} as the Kronecker product of R matrices:

$$\mathbf{K}_{zz} = \bigotimes\nolimits_{r=1}^{R} \mathbf{K}_{z_r z_r}, \qquad \Psi_{zz} \;= \bigotimes\nolimits_{r=1}^{R} \Psi_{z_r z_r} \tag{14}$$

where the functions used to construct $\mathbf{K}_{z_r z_r}$ and $\Psi_{z_r z_r}$ are the same as Equations 18 and 26 without the product over R. We must also give the covariance of the variational distributions, $\mathbf{S}_t$, Kronecker structure:

$$\mathbf{S}_t = \bigotimes\nolimits_{r=1}^{R} \mathbf{S}_{t,r} \tag{15}$$

We could also similarly structure the means, $\mathbf{m}_t$, however our implementation left these as general full vectors, since we can still exploit Kronecker structure when multiplying Kronecker matrices with full vectors. In fact, we never need to construct any full $M \times M$ matrix to compute the collapsed lower bound, nor any of its derivatives and instead we store each of the constituent matrices separately. For example, using straight-forward applications of the Kronecker matrix identities for inversion, multiplication and trace, we can compute the following term of Equation 11: $\mathrm{Tr}(\mathbf{K}_{zz}^{-1}\Psi_{zz})$, as

$$\mathrm{Tr}\left(\bigotimes\nolimits_{r=1}^{R} \mathbf{K}_{z_r z_r}^{-1} \Psi_{z_r z_r} \right) = \prod_r \mathrm{Tr}\left(\mathbf{K}_{z_r z_r}^{-1} \Psi_{z_r z_r} \right),$$

which only requires multiplication and inversion of M_r sized matrices. We also need to maintain Kronecker structure of the cross-kernel terms $\mathbf{k}_{zx} = \mathbf{k}_{xz}^\top$:

$$\mathbf{k}_{zx}^{(s,n)} = \bigotimes\nolimits_{r=1}^{R} \mathbf{k}_{z_r x_r}^{(s,n)}. \tag{16}$$

To allow efficient use of low-level matrix libraries, it is important to keep all the constituent vectors $\mathbf{k}_{z_r x_r}^{(s,n)}$ stacked together in contiguous memory as follows:

$$\mathbf{K}_{z_r x_r}^{(s,1:N_s)} = [\mathbf{k}_{z_r x_r}^{(s,1)}, \ldots, \mathbf{k}_{z_r x_r}^{(s,N_s)}]. \tag{17}$$

This allows us to compute, for example, $\mathbf{K}_{xz}^{(s,1:N_s)} \mathbf{K}_{zz}^{-1}$ using only R BLAS function calls rather than $N_s \times R$ calls separately.

There are two important consequences of this structuring. The first is that inducing points cannot be moved independently, we can only control the $z_r^{(m)}$ each of which controls the r^{th} co-ordinate of M/M_r inducing points in $\mathcal{Z}$. However, if we are content to place the inducing points on a grid and forgo the (computationally expensive) opportunity of optimising the locations, this distinction is moot. The other important consequence is the restriction of $\mathbf{S}_t$ to have Kronecker structure. Since this restricts the variational distribution $q(\mathbf{u}_t)$, the latter is necessarily less flexible than the full matrix equivalent. We can therefore expect the tightest variational bound achievable in the full matrix case will always be at least as good as in the Kronecker structured case; the latter may also induce a more complicated optimisation landscape.

3.5 Computational Complexity

The computational complexity of LPPA is a function of the total number of data points in all outputs $N = \sum_s N_s$, the number of latent functions T and the number of inducing points M (or for Kronecker structured kernel matrices $\max_r(M_r)$). As can be seen from Equation 8, the computational complexity is linear in N and T. The most significant computational costs are associated with inverting $M \times M$ (or $M_r \times M_r$) matrices, with complexity $\mathcal{O}(M^3)$ if computed via Gauss-Jordan elimination, and with matrix-matrix multiplications with complexity $\mathcal{O}(NM^2)$.

We note that both $\mathbf{K}_{zz}$ and Ψ_{zz} meet the requirements for inversion and matrix-vector multiplication using the Inverse Fast Multipole Method (IFMM) (Ambikasaran and Darve, 2014)—when computed using squared exponential kernel—which has $\mathcal{O}(n)$ complexity for both operations. However, the headline complexity is still governed by matrix-matrix multiplications of $\mathbf{S}$ in either case and is $\mathcal{O}(TNM^2)$.

3.6 Predictive Distribution

To evaluate the performance of LPPA we will compute a lower bound on the predictive log-likelihood, $\mathcal{L}_p \leq \log p(\mathscr{H}_{1:S}|\mathscr{D}_{1:S})$, of held out sources with $\mathscr{H}_s = \{\tilde{\mathbf{x}}^{(s,n)}\}_{n=1}^{\tilde{N}_s}$. The derivation of $\mathcal{L}_p$ begins by assuming the posterior distribution of the latent functions at the inducing points, $p(\mathbf{u}_{1:T}|\mathscr{D}_{1:S})$, is well approximated by the optimised variational distribution $q(\mathbf{u}_{1:T})$. The remaining steps follow the derivation of collapsed bound and the resulting bound is the same except there is no KL term.

When evaluating $\mathcal{L}_p$ there are two distinct use cases corresponding to whether we believe the held out data has the same rate as the training data, or whether they merely have the same latent functions f_t^2, albeit in different proportions. The former case corresponds to reusing the same learned output length scales, Γ, for test, and the latter corresponds to allowing Γ to be re-optimised, holding the remaining parameters fixed.

3.7 Model Identifiability

Like most factor models—NMF, LDA, etc.—LPPA is non-identifiable and non-unique; there may be multiple decompositions that are well supported by the data and the decomposition found is sensitive to initial conditions. The Gaussian process prior and Bayesian shrinkage will tend to reduce the non-identifiability of the solutions, by expressing a preference for smooth latent functions. Bayesian shrinkage will also tend to prevent over-fitting the number of latent functions.

The requirements for LPPA to generate unique solutions will be at least those for unique NMF solutions, i.e. complete factorial sampling of linearly independent basis (Donoho and Stodden, 2003).

4 EXPERIMENTS

In this section we will motivate and demonstrate the application of LPPA to several real world datasets.

We do not include experiments on synthetic data with known ground truth rate functions as they are not very informative. Due to non-identifiability the learned latent rate functions may not be very similar to the generative latent functions in a mean-squared-error sense. The learned rate functions are often sparse hybrids of the generative functions.

Approximate model recovery using LPPA usually requires a large number of points since the rate functions are only weakly observed, with each latent function dominating one of the outputs. However this is not representative of real world use on which we focus.

Neither model recovery nor identifiability is required for the model to be of practical value since the outputs are identifiable and, as we shall demonstrate, the ability of the model to generalise across data sets does improve predictive performance.

4.1 Benchmark

We benchmark against an algorithm combining Kernel Smoothing (KS) and Poisson-NMF (Lee and Seung, 2001) We first smooth each data set using truncated normal densities to construct a rate process, $\lambda_s(\mathbf{x}) = \sum_{n=1}^{N_s} \mathcal{N}_{\mathcal{X}}(\mathbf{x}; \mathbf{x}^{(s,n)}, \Sigma_s^*)$, for each data source s with diagonal covariances, Σ_s^* learned by leave-one-out cross validation (§A.7). Next we integrate the rate

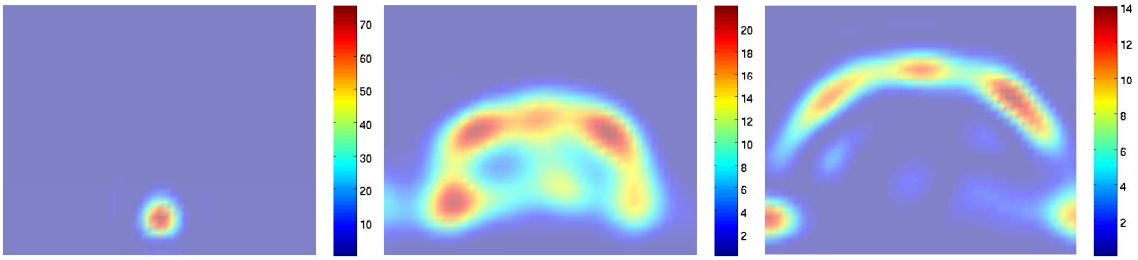

Figure 3: Bases computed by LPPA for the Basketball data set using 20 players. Note the sparsity of the three factors, and the fact that they strongly conform to three 'modes' of shot typically attempted by players. From left to right: the slam-dunk, the two-pointer shot, and finally the three-point boundary line and corners.

functions, λ_s, over D grid-cells, each of dimension $\Delta\mathbf{x}$, and denote l_{sd} as the result of the integral over the d^{th} cell. We then factorise the matrix $\mathbf{L} \in \mathbb{R}_+^{S\times D}$, with entries l_{sd}, using NMF as $\mathbf{L} \sim \mathcal{P}(\mathbf{AB})$, where $\mathbf{A} \in \mathbb{R}_+^{S\times T}$ is the so-called "activation" matrix, and $\mathbf{B} \in \mathbb{R}_+^{T\times D}$ is the "template" matrix. This two stage procedure results in a predictive rate function that is a positive weighted sum of piece-wise constant functions.

4.2 Twitter Data

Our first application is a simple 1-dimensional time series from Twitter. We pulled the Twitter streams of randomly selected politicians from Australia (19 politicians), the UK (24) and the US (25) during an 84-hour time window, specifically from midday on 1st June 2015 through to midnight on 5th June 2015.

We created 10 test/train splits of roughly equal size by dividing each stream in half (odd tweets randomly assigned). We then ran up to 1000 iterations of gradient decent on our LPPA model. Each split contained $\sim$ 672 tweets and each politician tweeted between 3 and 61 times (in each split). We used 10 random initialisations for LPPA and the benchmark, selecting the run with the best training likelihood; the latter was then used to compute the predictive likelihood. We repeated this for each split and averaged the result.

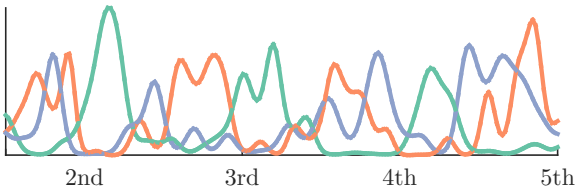

Figure 4: Bases computed by LPPA for Twitter data.

As shown in Figure 4, the three base functions inferred are periodic and it is clear that they do not fully overlap, the phase lag can be attributable to the 8-12 hour time difference between each of the three time-zones. The predictive log-likelihood performance[5] , Table 1, is very strong as compared to the competing method.

Table 1: Twitter held out predictive log-likelihoods.

LPPA ($\mathcal{L}_p$)	KS+NMF	IND. KS
-163.98	-379.77	-393.82

4.3 Basketball Data

For our next experiment we investigate an application analysed by Miller et al. (2014) and Gunter et al. (2014), specifically that of learning the scoring intensity of professional NBA Basketball players. Although players score from different locations there are several different key positions that influence where each player is likely to score from, as shown in Figure 3.

As described in Section 3.6 we can consider two use cases: 1) the common rate (CR) case, corresponds to the prediction of a held out set of shot data from players used to train the model and 2) the common topic (CT) case, corresponds the construction of rate processes for players not previously used to train the model using new output length scales, learnt for each new player independently.

As before we create test/train splits and use random restarts. Inducing points were fixed on an evenly spaced grid and we used a 25×30 grid for the benchmark. Experimental results[5] are given in Tables 2&3.

4.4 Wild Bird Data

In this experiment we used the wild-bird dataset previously investigated by Ioannis et al. (2012). The dataset

[5] Predictive performance can be improved by using the variational distribution as a Monte-Carlo integration proposal distribution as described in Lloyd et al. (2015a).

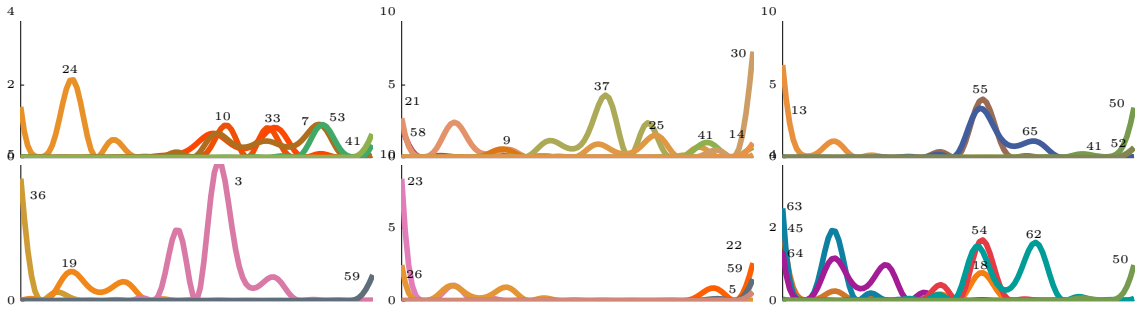

Figure 5: Bases computed by LPPA for the Wild Bird dataset. Each sub-plot represents the spatio-temporal distribution of putative communities of birds. The numbers and colour coding correspond those in Figure 6.

Table 2: Basketball held out predictive log-likelihoods.

	LPPA($\mathcal{L}_p$)		KS+NMF		IND. KS
	CR(1)	CT(2)	CR(1)	CT(2)	CR(1)
1	**-3599.2**	**-3577.7**	-3808.0	-4063.0	-4079.8
2	**-3701.4**	**-3602.8**	”	”	”
3	**-3648.3**	**-3610.9**	-3787.0	-3947.5	”
4	**-1866.6**	-1832.6	-1998.1	**-1806.5**	-2075.0
5	**-5647.8**	**-5657.4**	-6084.3	-6678.4	-6434.9
6	**-3503.9**	**-3523.3**	-3808.0	-4063.0	-4079.8

Table 3: Experimental Parameters.

	Kron	T	S	N_s	Inducing Pts
1	No	3	40	25	13×17
2	Yes*	3	40	25	13×17
3	No	4*	40	25	13×17
4	No	3	20*	25	13×17
5	No	3	40	50*	13×17
6	No	3	40	25	25×30*

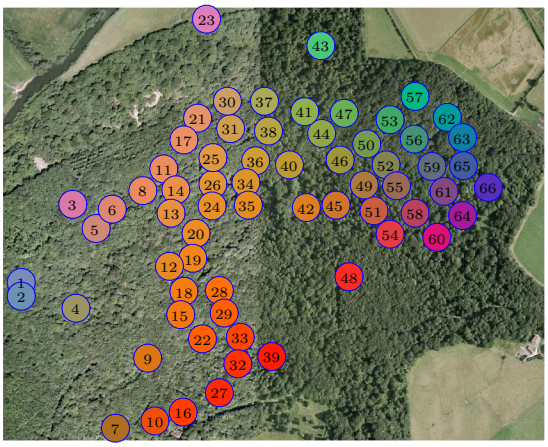

Figure 6: Feeding stations in Wytham Great Wood.

contains the times tagged wild birds arrive at a number of Radio Frequency Identification (RFID) equipped bird feeders distributed across Wytham Great Wood, near Oxford. The dataset contains hundreds of birds and hundreds of thousands of arrivals at dozens of locations shown Figure 6. We selected a subset of the data containing 14,742 arrivals by 274 birds at 37 locations over a 7 day period.

We model each bird as a separate output and latent functions are defined over a mixed continuous discrete co-ordinate space consisting of arrival time and feeder ID. Although the location identifiers are discrete, the kernel between feeders reflect their geographic proximity.[6] We used six of these continuous-discrete latent functions, which are shown in Figure 5.

Factorisation reveals likely communities of birds, each of which has a distinct arrival intensity for each feeder. We found that most birds attached strongly to one community, suggesting that much of the community structure has been captured.

5 CONCLUSION

We have presented a Bayesian factor model for continuous Poisson process intensities and an associated variational inference algorithm. The approach yields sparse, smooth and interpretable latent factors. We have demonstrated the model on real world datasets.

This work has many possible extensions and applications, for example modelling dynamic interaction networks. The latter arise when people or computers communicate, creating ephemeral links that can be thought of as point processes. We may wish to infer shared structure between activity on each link that arises from the community structure of the senders and receivers. LPPA can easily be adapted for this purpose (see Lloyd et al. (2015b) and §A.9).

[6]The presence of this discrete dimension means that one of the integrals used to compute Equation 26 becomes a finite sum, see §A.8.

Acknowledgements

The authors would like to thank James Hensman for helpful discussions. Chris Lloyd is funded by a DSTL National PhD Scheme Studentship. Tom Gunter is supported by UK Research Councils. Tom Nickson is supported by an EPSRC/ORCHID PhD Studentship.

References

R. P. Adams, I. Murray, and D. J. C. MacKay. Tractable Nonparametric Bayesian Inference in Poisson Processes with Gaussian Process Intensities. In *ICML*, 2009.

S. Ambikasaran and E. Darve. The inverse fast multipole method. *arXiv pre-print 1407.1572*, 2014.

L. U. Ancarani and G. Gasaneo. Derivatives of any order of the confluent hypergeometric function ${}_1F_1(a, b, z)$ with respect to the parameter a or b. *Journal of Mathematical Physics*, 49(6), 2008.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

A. T. Cemgil. Bayesian Inference for Non-negative Matrix Factorisation Models. *Computational Intelligence and Neuroscience*, 2009.

David Donoho and Victoria Stodden. When Does Non-Negative Matrix Factorization Give Correct Decomposition into Parts? In *NIPS*, 2003.

N. Eisenbaum and H. Kaspi. On permanental processes. *Stochastic Processes and their Applications*, 119(5):1401 – 1415, 2009.

S. Flaxman, A. Wilson, D. Neill, H. Nickisch, and A. Smola. Fast Kronecker Inference in Gaussian Processes with non-Gaussian Likelihoods. In *ICML*, 2015.

T. Gunter, C. Lloyd, M. A. Osborne, and S. J. Roberts. Efficient Bayesian Nonparametric Modelling of Structured Point Processes. In *UAI*, 2014.

J. Hensman, M. Rattray, and N. D. Lawrence. Fast Variational Inference in the Conjugate Exponential Family. In *NIPS*, 2012.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian Processes for Big Data. In *UAI*, 2013.

J. B. Hough, M. Krishnapur, Y. Peres, and B. Virg. Determinantal processes and independence. *Probability Surveys*, 2006.

P. Ioannis, S. J. Roberts, I. Rezek, and B. C. Sheldon. Inferring social network structure in ecological systems from spatio-temporal data streams. *Journal of The Royal Society Interface*, 9(76):3055–3066, 2012.

T. Iwata, A. Shah, and Z. Ghahramani. Discovering Latent Influence in Online Social Activities via Shared Cascade Poisson Processes. In *KDD*, 2013.

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

Y. L. Kom Samo and S. J. Roberts. Scalable Nonparametric Bayesian Inference on Point Processes with Gaussian Processes. In *ICML*, 2015.

D. D. Lee and S. H. Seung. Algorithms for Non-negative Matrix Factorization. In *NIPS*, 2001.

W. Lian, V. Rao, B. Eriksson, and L. Carin. Modeling Correlated Arrival Events with Latent Semi-Markov Processes. In *ICML*, 2014.

S. Linderman and R. P. Adams. Discovering Latent Network Structure in Point Process Data. In *ICML*, 2014.

C. Lloyd, T. Gunter, M. A. Osborne, and S. J. Roberts. Variational Inference for Gaussian Process Modulated Point Processes. In *ICML*, 2015a.

C. Lloyd, T. Gunter, M. A. Osborne, and S. J. Roberts. Inferring Dynamic Interaction Networks with N-LPPA. In *Networks in the social and information sciences, NIPS workshop*, 2015b.

A. Matthews, J. Hensman, R. Turner, and Z. Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In *AISTATS*, 2016.

A. Miller, L. Bornn, R. P. Adams, and K. Goldsberry. Factorized Point Process Intensities: A Spatial Analysis of Professional Basketball. In *ICML*, 2014.

J. Moller, A. R. Syversveen, and R. P. Waagepetersen. Log Gaussian Cox Processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2nd edition, 2006.

M. N. Schmidt and M. Morup. Nonparametric Bayesian Modeling of Complex Networks: an Introduction. *IEEE Signal Processing Magazine*, 2013.

A. Simma and M. I. Jordan. Modeling Events with Cascades of Poisson Processes. In *UAI*, 2010.

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.

A. Wilson, E. Gilboa, A. Nehorai, and J. Cunningham. Fast kernel learning for multidimensional pattern extrapolation. In *NIPS*, 2014.