Multi-task Gaussian process modelling for point process data

Virginia Aglietti

University of Warwick, The Alan Turing Institute

GP Summer School, September 2020

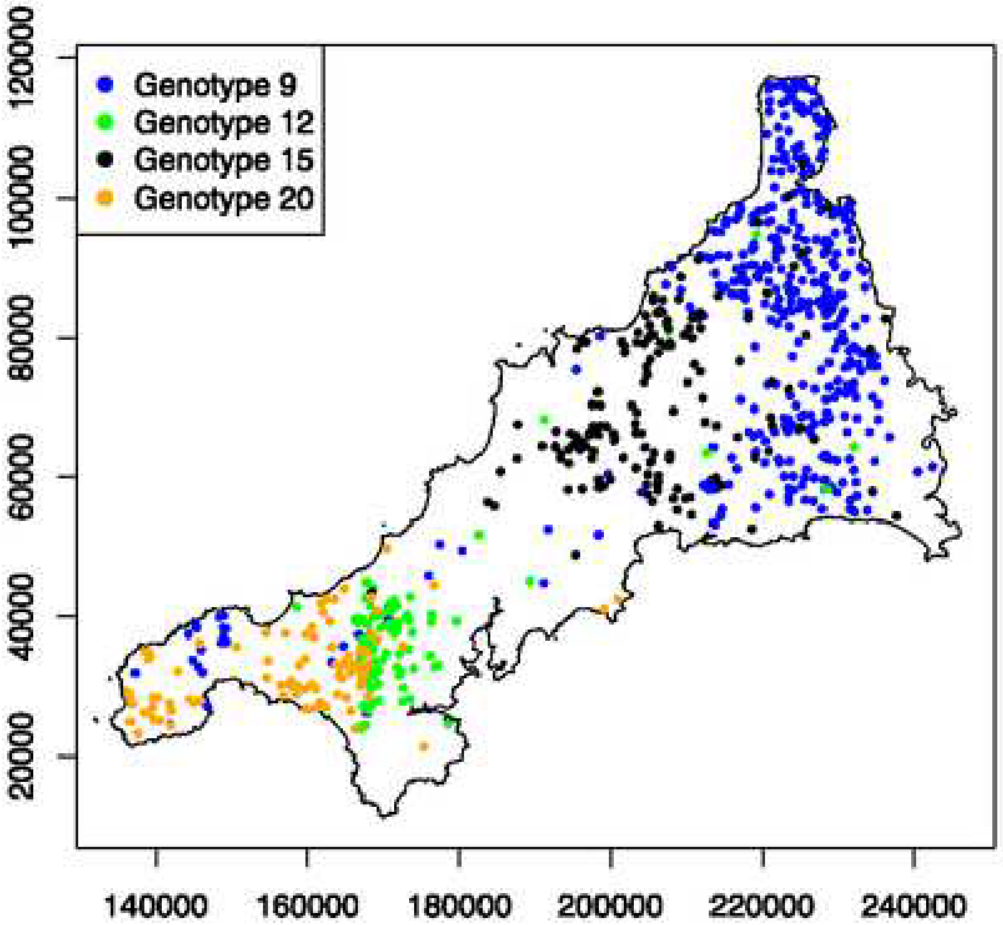
Outline

Motivation

Many social processes can be seen as point processes.

Motivation

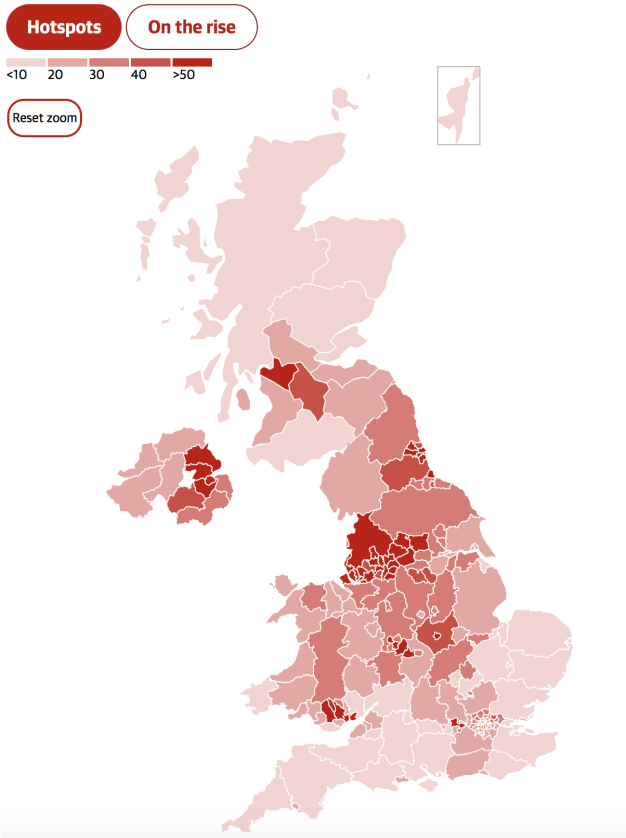
Many social processes can be seen as point processes.



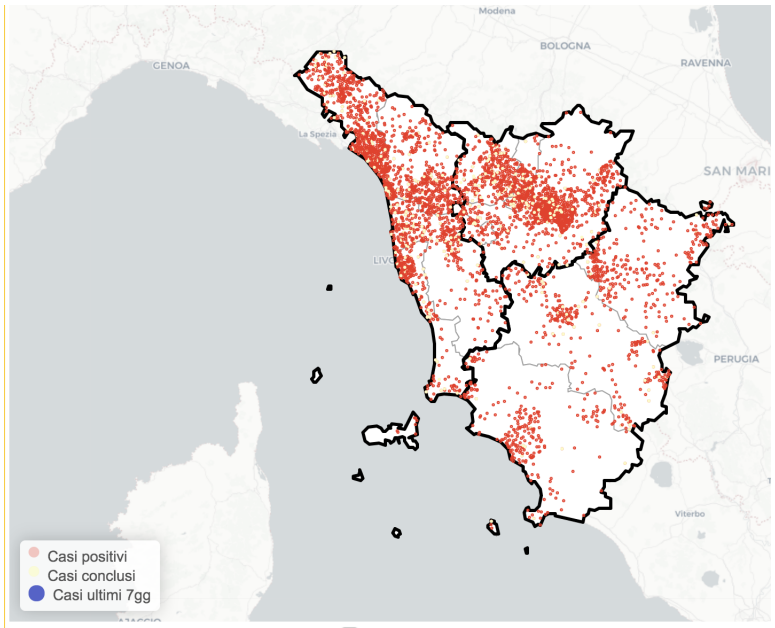
- Cases of bovine tuberculosis (BTB) over the years 1989–2002 in Cornwall, UK.
- Four different strains of the disease.
- "Nonparametric estimation of spatial segregation in a multivariate point process: bovine tuberculosis in Cornwall, UK." by Diggle P. et al. (2005).

Motivation

Many social processes can be seen as point processes.



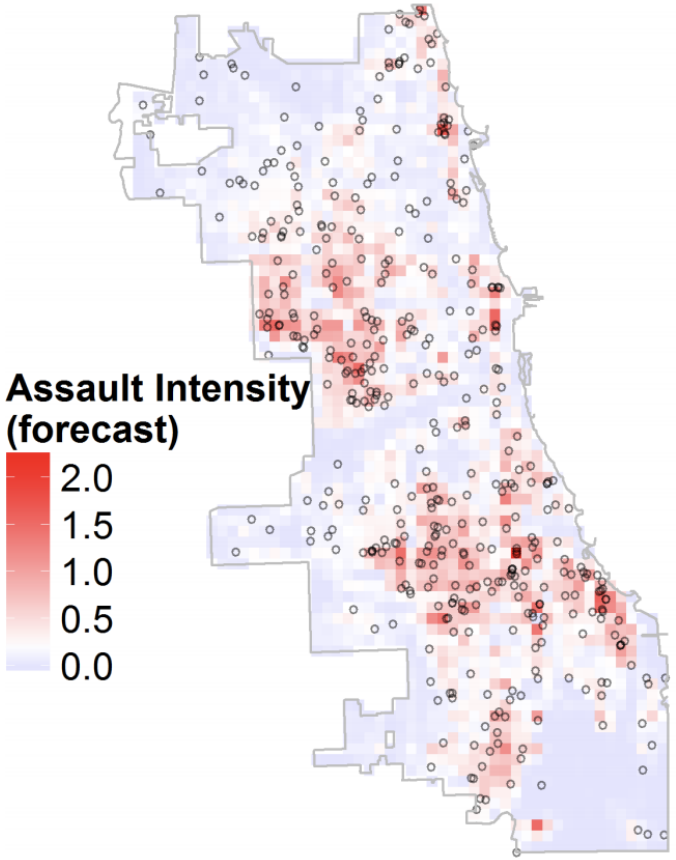
- Counts of Covid19 events in UK



- Covid19 events in Tuscany

Motivation

Many social processes can be seen as point processes.



- Cases of "assaults" over the year 2004-2013 in Chicago, USA.
- "Fast Kronecker Inference in Gaussian Processes with non-Gaussian Likelihoods" by Flaxman S. et al. (2015).

Motivation

Many social processes can be seen as point processes.

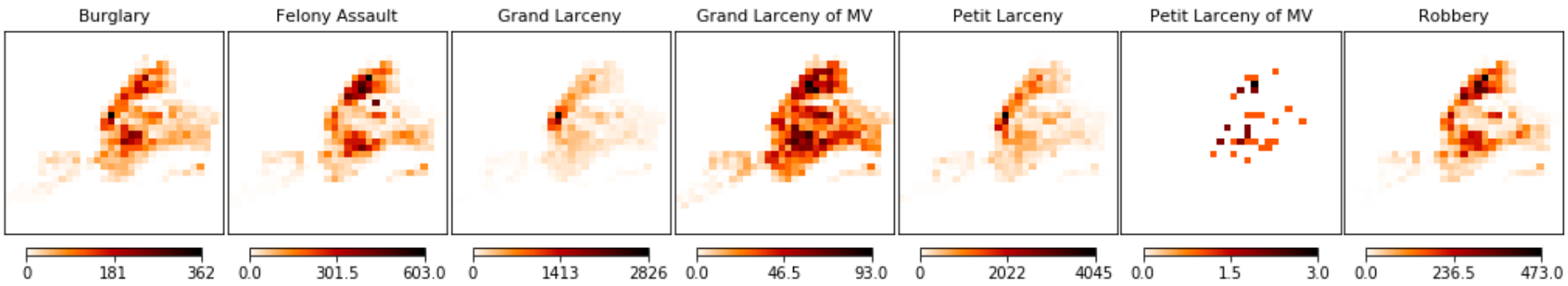


Figure 1: Event counts for seven different crimes reported in 2016 in New York City, USA.

Motivation

These social processes are characterized by:

- Spatial and temporal correlation;
- Correlation structure that changes in time and space (non-stationary) in a dependent manner (non-separable);
- Cross-correlation

Goal: Develop a scalable algorithm capable of jointly modelling social processes accounting for their complexities.

Cox processes

Estimating the intensity rate of events over a continuous space is a common problem for real-world applications.

Doubly stochastic Poisson process or **Cox process** [Cox 1955]:

- observed events are assumed to be generated from a Poisson process
- intensity function is modelled as another random process with a given prior probability measure

The LGCP model

The Log-Gaussian Cox Process (LGCP) [Møller et al., 1998] is an **inhomogeneous Poisson process** with a **stochastic intensity** function:

$$y_A|\lambda(\mathbf{x}) \sim \text{Poisson}\left(\int_{\mathbf{x}\in A} \lambda(\mathbf{x})d\mathbf{x}\right),$$

where:

$$\lambda(\mathbf{x}) = \exp\{f(\mathbf{x})\}$$
$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}))$$

The LGCP model

The Log-Gaussian Cox Process (LGCP) [Møller et al., 1998] is an **inhomogeneous Poisson process** with a **stochastic intensity** function:

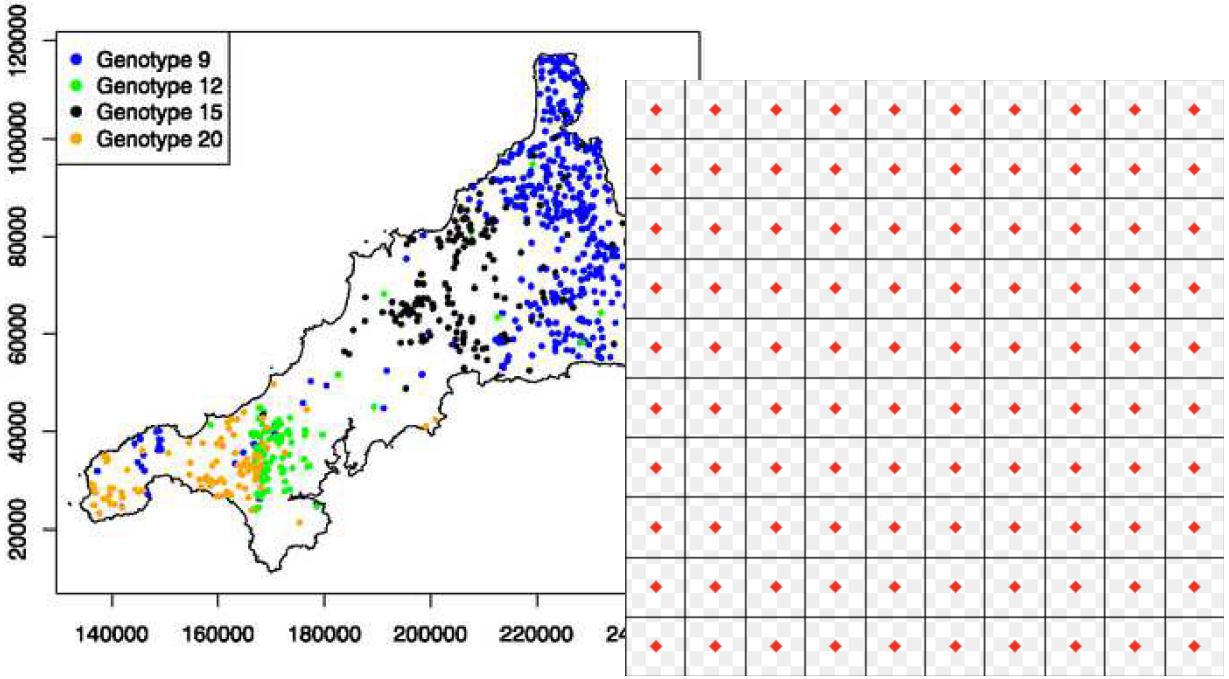
$$y_A|\lambda(\mathbf{x}) \sim \text{Poisson}\left(\int_{\mathbf{x}\in A} \lambda(\mathbf{x})d\mathbf{x}\right)$$

The LGCP model

The Log-Gaussian Cox Process (LGCP) [Møller et al., 1998] is an **inhomogeneous Poisson process** with a **stochastic intensity** function:

$$y_A|\lambda(\mathbf{x}) \sim \text{Poisson}\left(\int_{\mathbf{x}\in A} \lambda(\mathbf{x})d\mathbf{x}\right)$$

Computational grid for tractable inference:



The LGCP model

The Log-Gaussian Cox Process (LGCP) [Møller et al., 1998] is an **inhomogeneous Poisson process** with a **stochastic intensity** function:

$$y_A|\lambda(\mathbf{x}) \sim \text{Poisson}\left(\int_{\mathbf{x}\in A} \lambda(\mathbf{x})d\mathbf{x}\right)$$

Computational grid for tractable inference:

$$y_n|\lambda(\mathbf{x}) \sim \text{Poisson}\left(\lambda(\mathbf{x}_n)\right)$$

Limitations of the LGCP model

1. Computational grid
2. Single-task model
3. High computational cost due to the use of expensive MCMC schemes

Limitations of the LGCP model

1. Computational grid $\rightarrow$ *"Structured variational inference in continuous cox process models." by Aglietti et al. (2019)*
2. Single-task model
3. High computational cost when using expensive MCMC schemes

The LGCP model

1. Computational grid
2. Single-task model $\rightarrow$ Multi-task model
3. High computational cost $\rightarrow$ Variational Inference scheme

Motivation

Event counts for different types of crime are highly correlated.

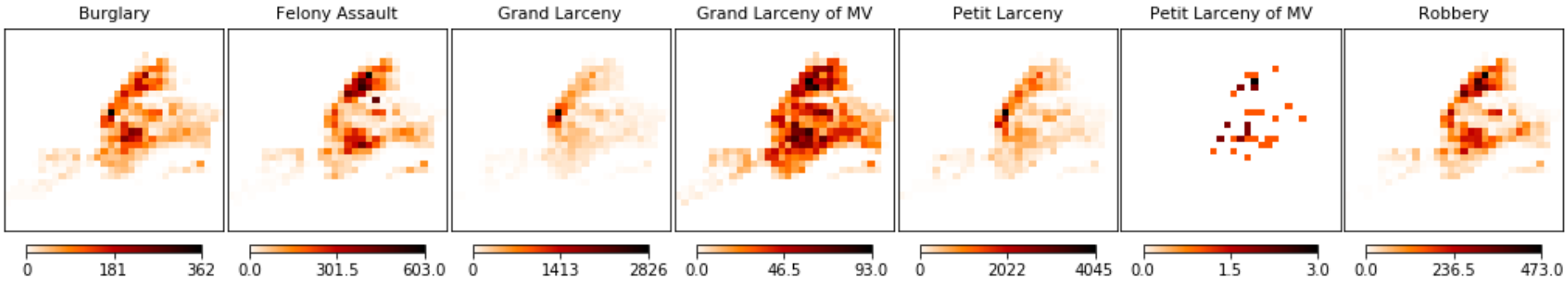
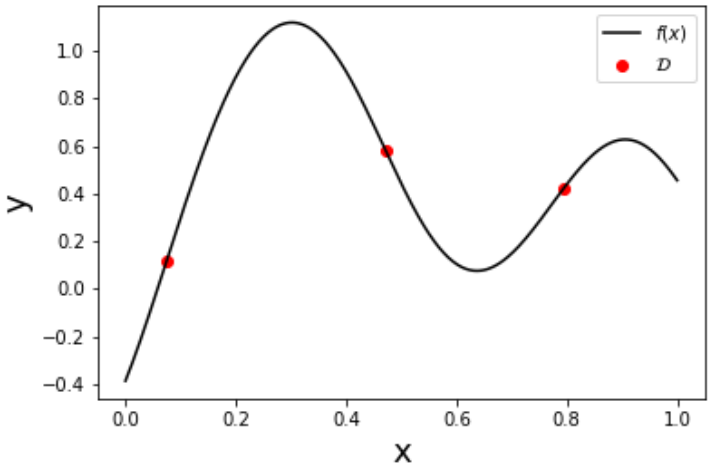


Figure 2: Event counts for seven different crimes reported in 2016 in New York City, USA.

$\rightarrow$ Learn a model that capture the dependencies between these processes

Single-task GP regression



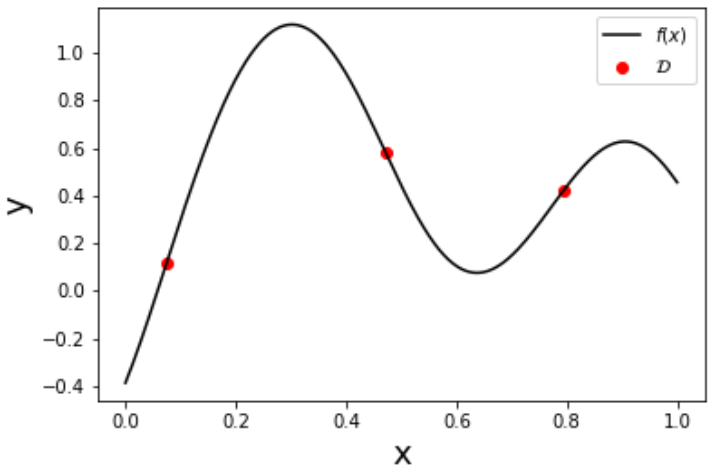
GP regression:

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$
$$y_i = f(\mathbf{x}_i) + \epsilon_i \quad \epsilon_i \sim N(0, \sigma^2)$$
$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \quad N = 3$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} + \sigma^2 I \right)$$

with $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Single-task GP regression

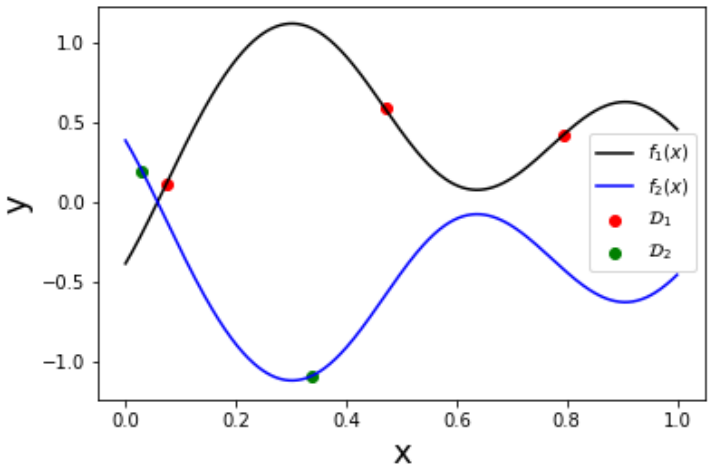


$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$
$$y_i = f(\mathbf{x}_i) + \epsilon_i \quad \epsilon_i \sim N(0, \sigma^2)$$
$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \quad N = 3$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \underbrace{\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}}_{K} + \sigma^2 I \right)$$

We can get the prediction for a new test point $\mathbf{x}^*$ by computing $p(f(\mathbf{x}^*)|\mathbf{y})$ in closed form.

Multi-task GP regression



$$f_1(\mathbf{x}) \sim \mathcal{GP}(0, k_1(\mathbf{x}, \mathbf{x}'))$$
$$f_2(\mathbf{x}) \sim \mathcal{GP}(0, k_2(\mathbf{x}, \mathbf{x}'))$$
$$\mathcal{D}_1 = \{(\mathbf{x}_i^1, y_i^1)\}_{i=1}^{N_1} \quad N_1 = 3$$
$$\mathcal{D}_2 = \{(\mathbf{x}_i^2, y_i^2)\}_{i=1}^{N_2} \quad N_2 = 2$$
$$y_i^1 = f_1(\mathbf{x}_i^1) + \epsilon_i \quad \epsilon_i \sim N(0, \sigma_1^2)$$
$$y_i^2 = f_2(\mathbf{x}_i^2) + \epsilon_i \quad \epsilon_i \sim N(0, \sigma_2^2)$$

$$\begin{bmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_1 & \mathbf{0} \\ \mathbf{0} & K_2 \end{bmatrix} + \begin{bmatrix} \sigma_1^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_2^2 \mathbf{I} \end{bmatrix} \right)$$

Intrinsic coregionalization model (ICM)

We assume the two functions to be generated from latent process:

$$u(\mathbf{x}) \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$$
$$f_1(\mathbf{x}) = w_1 u(\mathbf{x})$$
$$f_2(\mathbf{x}) = w_2 u(\mathbf{x})$$

$$\begin{bmatrix} \mathbf{y}^1 \\ \mathbf{y}^2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \mathbf{K} + \begin{bmatrix} \sigma_1^2 \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \sigma_2^2 \mathbf{I} \end{bmatrix} \right)$$

$$\mathbf{K} = \mathbf{B} k(\mathbf{x}, \mathbf{x}') = \mathbf{w}\mathbf{w}^T k(\mathbf{x}, \mathbf{x}') \quad \mathbf{w} = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^T$$

Intrinsic coregionalization model (ICM)

In general we have a set of functions $\{f_p(\mathbf{x})\}_{p=1}^{P}$ and we assume

$$f_p(\mathbf{x}) = \sum_{i=1}^{R} w_p^i u^i(\mathbf{x})$$

with $\{u^i(\mathbf{x})\}_{i=1}^{R}$ are independent GPs with the same $k(\mathbf{x}, \mathbf{x}')$.

Define $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), ..., f_P(\mathbf{x})]^T$ then we have:

$$\text{cov}(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}')) = \mathbf{B}k(\mathbf{x}, \mathbf{x}')$$

Other multi-task GP models

- Semiparametric Latent factor model (SLFM)
- Linear Coregionalization model (LCM)
- Process convolutions
- ...

A multi-task LGCP model

The **multivariate** LGCP (MLGCP) [Diggle et al., 2013] considers P types of points with an intensity given by:

$$\lambda_p(\mathbf{x}) = \exp(\beta + f_0(\mathbf{x}) + f_p(\mathbf{x}))$$

where $f_0(\mathbf{x})$ indicates a GP common to all tasks while $f_p(\mathbf{x})$ denotes a GP specific to the point process of type p. Inference proceeds via a MALA algorithm.

Coregionalisation models for point processes

The intensity function is a linear combination of Q independent Gaussian processes:

$$\lambda_p(\mathbf{x}) = \exp(\sum_{q=1}^{Q} \underbrace{w_{pq}}_{\textit{deterministic}} \quad u_q(\mathbf{x})).$$

The covariance $\text{Cov}[f_p(\mathbf{x}), f_{p'}(\mathbf{x}')]$ is given by $K(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} B_q k_q(\mathbf{x}, \mathbf{x}')$ where B_q is known as a coregionalisation matrix.

A Bayesian treatment has been proposed by Schmidt & Gelfand (2003). The weight parameters have an inverse Wishart prior and inference proceeds with MCMC.

Modelling goals

- increase modelling flexibility with respect to MLGCP and ICM
- develop a full Bayesian model that propagates uncertainty in the mixing weights

Multi-task Cox process model (MCPM)

Latent functions:

- Q **uncorrelated** $\mathcal{GP}$:

$$p(\mathbf{F}|\boldsymbol{\theta}) = \prod_{q=1}^{Q} p(\mathbf{F}_{\bullet q}|\boldsymbol{\theta}_q)$$
$$= \prod_{q=1}^{Q} \mathcal{N}(\mathbf{F}_{\bullet q}|\mathbf{0}, \mathbf{K}_{xx}^{q}),$$

where $\boldsymbol{\theta}_q$ are the hyper-parameters for the q-th latent function.

Mixing weights:

- **Coupled weights**, independent across latent processes:

$$p(\mathbf{W}|\boldsymbol{\theta}_w) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{W}_{\bullet q}; \mathbf{0}, \mathbf{K}_{w}^{q}),$$

where $\boldsymbol{\theta}_w$ denotes the hyper-parameters.

Graphical model representation

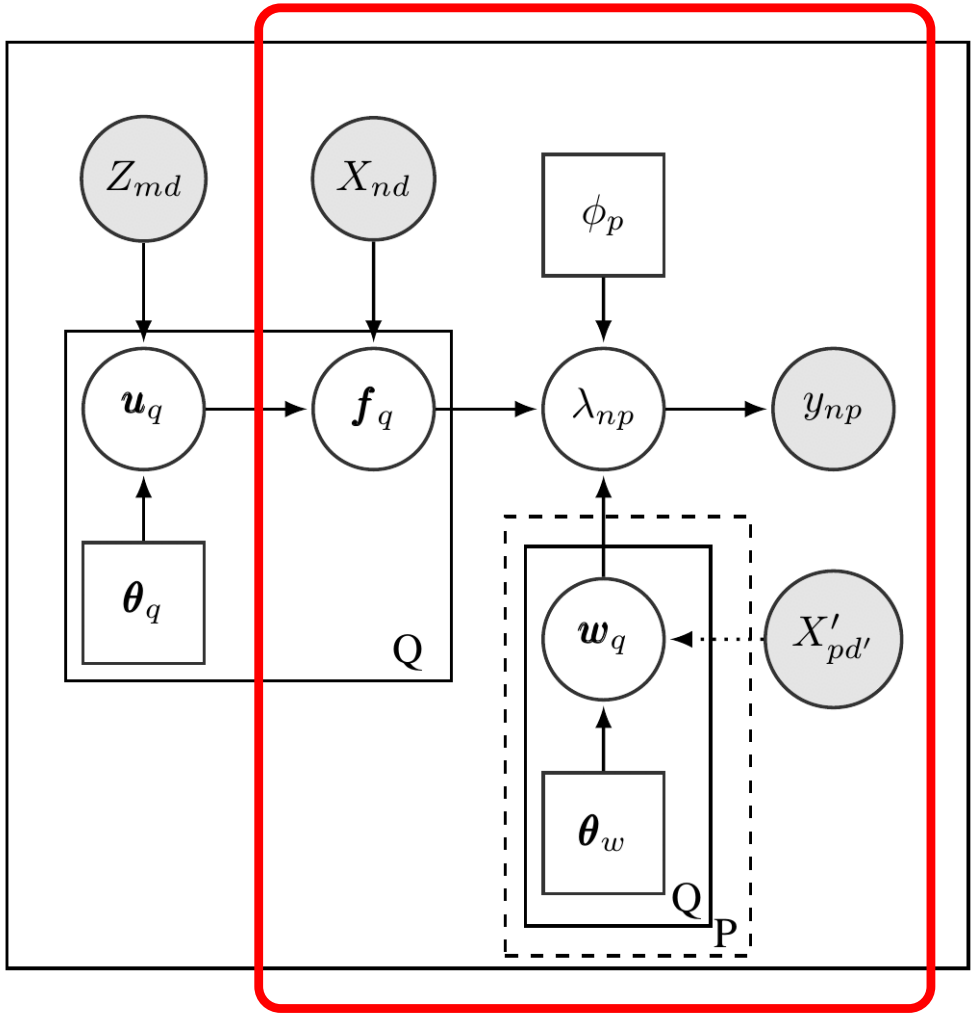


Figure 3: MCPM- Plate diagram. $X'_{pd'}$ represents the inputs for the GP prior on $\mathbf{W}$. When placing a Normal prior on each w_{pq}, we introduce the additional factorization across P (dashed plate).

Other models

- **Sigmoidal transformation**: Adams et al. (2009), Gunter et al. (2014), Lian et al. (2015), Donner and Opper (2018).
- **Square transformation**: Walder and Bishop (2017), Lloyd et al. (2015), Lian et al. (2015), Lloyd et al (2016), John and Hensman (2018).

Inference in GP modulated point process models

GP regression:

- Gaussian prior :
  $f(\mathbf{x}) \sim \mathcal{N}(0, k(\mathbf{x}, \mathbf{x}')) = p(f)$
- Gaussian likelihood: $y \sim \mathcal{N}(f, \sigma^2 I)$
- Gaussian posterior:
  $p(f|y) \propto \mathcal{N}(y|f, \sigma^2 I)\mathcal{N}(f|0, \sigma^2 I)$

PP model:

- Gaussian prior :
  $f(\mathbf{x}) \sim \mathcal{N}(0, k(\mathbf{x}, \mathbf{x}')) = p(f)$
- **Non** Gaussian likelihood:
  $y \sim \text{Poisson}(\exp(f))$
- **Non** Gaussian posterior:
  $p(f|y) = \frac{p(f)(y|f)}{\int p(f)(y|f)df}$

Inference in GP modulated point process models

GP regression:

- Gaussian prior : $f(\mathbf{x}) \sim \mathcal{N}(0, k(\mathbf{x}, \mathbf{x}')) = p(f)$
- Gaussian likelihood: $y \sim \mathcal{N}(f, \sigma^2 I)$
- Gaussian posterior: $p(f|y) \propto \mathcal{N}(y|f, \sigma^2 I)\mathcal{N}(f|0, \sigma^2 I)$

PP model:

- Gaussian prior : $f(\mathbf{x}) \sim \mathcal{N}(0, k(\mathbf{x}, \mathbf{x}')) = p(f)$
- **Non** Gaussian likelihood: $y \sim \text{Poisson}(\exp(f))$
- **Non** Gaussian posterior: $p(f|y) = \frac{p(f)(y|f)}{\int p(f)(y|f)df}$

- Sampling methods to obtain samples from the posterior
- Approximation of the posterior with something of known form $\rightarrow$ Variational inference

Variational inference

Idea: Minimize a divergence between the true posterior distribution $p(f|y)$ and an approximate posterior $q(f)$ of known form e.g. $q(f) = \mathcal{N}(\mu, \mathbf{C})$

$$KL(q(f)||p(f|y)) = \int q(f) \log \frac{q(f)}{p(f|y)} df$$

Adjust the variational parameters of $q(f)$ to minimize the KL divergence.

Variational inference

$$KL(q(f)||p(f|y)) = -\underbrace{(\mathbb{E}_{q(f)}[\log(f,y)] - \mathbb{E}_{q(f)}[\log(q(f))])}_{\text{ELBO}} + \log p(y)$$

Minimizing the KL divergence is equivalent to maximizing the ELBO which is a lower bound on the log-evidence.

Inference goals for the MCPM model

- Use a VI algorithm
- Reduce the computational cost, especially when P is large
- Avoid mixing issues of MCMC methods
- Get rid of further Monte Carlo approximations in the optimisation procedure

Scalable Variational Inference

We introduce an **augmented prior over the latent functions** [Titsias, 2009 and Bonilla et al., 2016] defined by:

$$p(\mathbf{U}|\boldsymbol{\theta}) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{U}_{\bullet q}; \mathbf{0}, \mathbf{K}_{zz}^{q}) \quad \text{and} \quad p(\mathbf{F}|\mathbf{U}, \boldsymbol{\theta}) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{F}_{\bullet q}; \tilde{\boldsymbol{\mu}}_q, \widetilde{\mathbf{K}}^{q}),$$

where $\tilde{\boldsymbol{\mu}}_q = \mathbf{K}_{xz}^{q}(\mathbf{K}_{zz}^{q})^{-1}\mathbf{U}_{\bullet q}$ and $\widetilde{\mathbf{K}}^{q} = \mathbf{K}_{xx}^{q} - \mathbf{K}_{xz}^{q}(\mathbf{K}_{zz}^{q})^{-1}\mathbf{K}_{zx}^{q}$.

$\mathbf{U}_{\bullet q}$ denotes the inducing process for $\mathbf{F}_{\bullet q}$ computed in the $M \times D$ matrix $\mathbf{Z}_q$ of *inducing inputs* ($M \ll N$).

Scalable Variational Inference

Idea: Minimise the *distance* between the true posterior distribution and the approximate posterior distribution, the variational distribution.

$$\underbrace{p(\mathbf{F},\mathbf{U},\mathbf{W}|\mathcal{D})}_{\text{True posterior distribution}} \rightarrow \underbrace{q(\mathbf{F},\mathbf{U},\mathbf{W}|\boldsymbol{\nu})}_{\text{Variational distribution}} = p(\mathbf{F}|\mathbf{U})\,\underbrace{q(\mathbf{U}|\boldsymbol{\nu}_u)}_{(1)}\,\underbrace{q(\mathbf{W}|\boldsymbol{\nu}_w)}_{(2)}$$

(1) $q(\mathbf{U}|\boldsymbol{\nu}_u) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{U}_{\bullet q}; \mathbf{m}_q, \mathbf{S}_q)$

(2) $q(\mathbf{W}|\boldsymbol{\nu}_w) = \prod_{q=1}^{Q} \mathcal{N}(\mathbf{W}_{\bullet q}; \boldsymbol{\omega}_q, \boldsymbol{\Omega}_q)$

where $\boldsymbol{\nu}_u = \{\mathbf{m}_q, \mathbf{S}_q\}$ are the variational parameters.

Scalable Variational Inference

Minimization of the KL divergence $\rightarrow$ Maximisation of the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{elbo}}(\boldsymbol{\nu}) = \mathcal{L}_{\text{kl}}(\boldsymbol{\nu}) + \mathcal{L}_{\text{ell}}(\boldsymbol{\nu}), \tag{1}$$

$$\mathcal{L}_{\text{kl}}(\boldsymbol{\nu}) = -\text{KL}(q(\mathbf{F}, \mathbf{U}, \mathbf{W}|\boldsymbol{\nu}) \| p(\mathbf{F}, \mathbf{U}, \mathbf{W})) \tag{2}$$

$$= -\text{KL}(q(\mathbf{U}|\boldsymbol{\nu}_u) \| p(\mathbf{U})) - \text{KL}(q(\mathbf{W}|\boldsymbol{\nu}_w) \| p(\mathbf{W})), \tag{3}$$

$$\mathcal{L}_{\text{ell}}(\boldsymbol{\nu}) = \underbrace{\mathbb{E}_{q(\mathbf{F},\mathbf{U},\mathbf{W}|\boldsymbol{\nu})}[\log p(\mathbf{Y}|\mathbf{F}, \mathbf{W})]}_{\text{Can we compute this expectation in closed form?}} \tag{4}$$

$$= -\sum_{n=1}^{N}\sum_{p=1}^{P} \exp(\phi_p)\mathbb{E}_{q(\mathbf{F}_{n\bullet})q(\mathbf{W}_{p\bullet})}\left(\exp\left(\mathbf{W}_{p\bullet}\mathbf{F}_{n\bullet}\right)\right) \tag{5}$$

$$+ \sum_{n=1}^{N}\sum_{p=1}^{P}\sum_{q=1}^{Q} [y_{np}(\omega_{pq}\mu_{nq} + \phi_p) - \log(y_{np}!)] \tag{6}$$

Scalable Variational Inference

$$\mathbb{E}\left[\exp\left(t\mathbf{W}_{p\bullet}\mathbf{F}_{n\bullet}\right)\right] = \mathrm{MGF}_{\mathbf{W}_{p\bullet}\mathbf{F}_{n\bullet}}(t) \tag{7}$$

where $\mathrm{MGF}_{\mathbf{W}_{p\bullet}\mathbf{F}_{n\bullet}}(t)$ denotes the moment generating function of $\mathbf{W}_{p\bullet}\mathbf{F}_{n\bullet}$ in t. The random variable $\mathbf{W}_{p\bullet}\mathbf{F}_{n\bullet}$ is the sum of products of independent Gaussians and its MGF is thus given by:

$$\mathrm{MGF}_{\mathbf{W}_{p\bullet}\mathbf{F}_{n\bullet}}(t) = \prod_{q=1}^{Q} \frac{\exp\left[\frac{t\gamma_{pq}\tilde{\mu}_{nq} + \frac{1}{2}(\tilde{\mu}_{nq}^2 K_w^{qp} + \gamma_{pq}^2 \widetilde{K}^{qn})t^2}{1 - t^2 K_w^{qp}\widetilde{K}^{qn}}\right]}{\sqrt{1 - t^2 K_w^{qp}\widetilde{K}^{qn}}}, \tag{8}$$

where the expectation is computed with respect to the prior distribution of $\mathbf{W}_{p\bullet}$ and $\mathbf{F}_{n\bullet}$; γ_{pq} is the prior mean of w_{pq}; $\widetilde{K}^{qn}$ denotes the variance of f_{nq}; and K_w^{qp} is the variance of w_{pq}.

Computational complexity

The time complexity of the algorithm is of order $\mathcal{O}(M^3)$.

- The KL-divergence includes distributions over M-dimensional variables and P-dimensional variables ($M \gg P$). Its computational complexity is thus independent of N.

- The ELL term decomposes as a sum of expectations over N. This enables the use of stochastic optimization techniques thus also making this term independent of N.

Experimental comparison

Baselines:

- Variational LGCP model
- Variational formulation of ICM with Poisson likelihood implemented in GPflow

Performance measures:

- Root mean square error (RMSE)
- Negative log predicted likelihood (NLPL)
- Empirical coverage of the posterior counts distribution (EC)

Setting:

- We show transfer by partitioning the spatial extent in subregions and create missing data folds for each task.

Synthetic data experiment

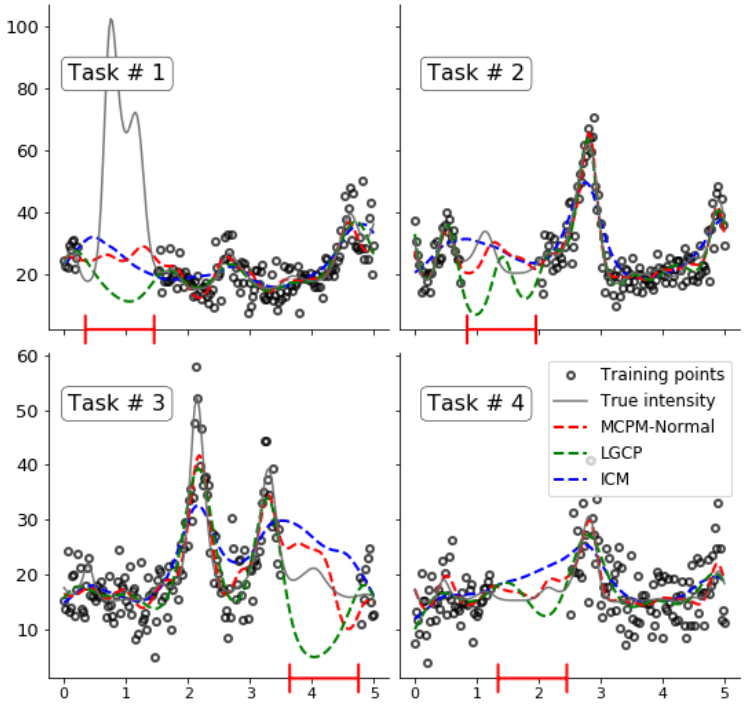


Figure 4: Four related tasks evaluated at 200 evenly spaced points in $[-1.5, 1.5]$. The red intervals denote the 50 contiguous observations removed from the training set of each task.

	RMSE				NLPL				CPU
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**	time
MCPM-N	38.61	7.86	5.71	**4.68**	20.99	3.75	**3.31**	**3.02**	**0.18**
MCPM-GP	**38.58**	**7.69**	**5.70**	4.71	**20.95**	**3.70**	**3.31**	3.03	0.25
LGCP	48.17	14.32	11.83	5.38	43.40	8.78	8.98	3.27	0.32
ICM	39.07	7.96	7.88	6.03	21.81	3.76	3.77	3.38	0.52

	Empirical Coverage (EC)			
	1	**2**	**3**	**4**
MCPM-N	0.80/0.12	**0.99**/0.58	0.92/0.57	**0.94**/**0.83**
MCPM-GP	**0.95**/**0.19**	0.72/**0.67**	**1.00**/**0.78**	0.92/0.75
ICM	0.75/0.03	0.66/0.60	0.62/0.50	0.93/0.42

Table 1: *Above:* Performance on the missing intervals. MCPM-N and MCPM-GP denote independent and correlated prior respectively. Time in seconds per epoch. Lower values of NLPL are better. *Below:* In-sample/Out-of-sample 90% CI coverage for the predicted event counts distributions. Higher values of EC are better.

Transfer experiment - CRIME dataset

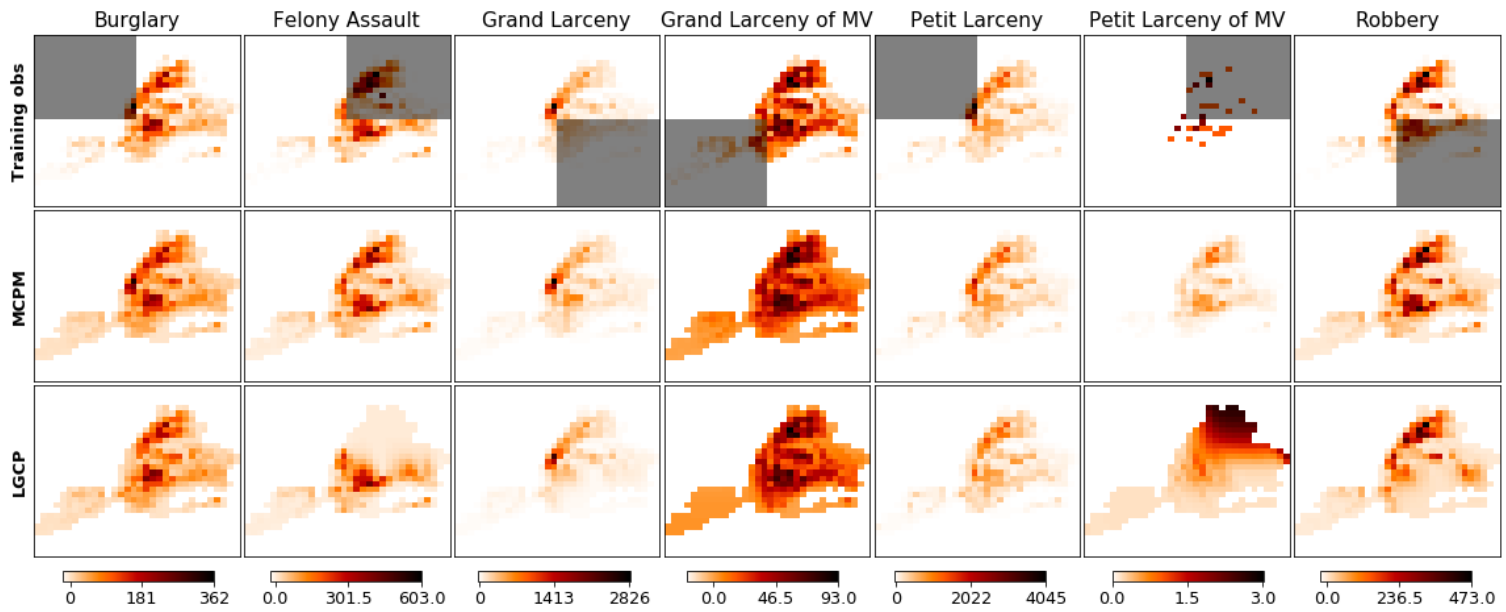


Figure 5: Estimated intensity surface when introducing missing data regions.

	Standardized NLPL (per cell)							CPU time
	1	**2**	**3**	**4**	**5**	**6**	**7**	
MCPM-N	**0.56** (0.10)	0.91 (0.27)	**0.66** (0.30)	**1.09** (0.27)	0.85 (0.52)	**10.29** (2.51)	**0.42** (0.05)	**2.85**
MCPM-GP	0.72 (0.18)	**0.75** (0.18)	0.94 (0.55)	1.53 (0.52)	**0.57** (0.19)	18.76 (8.25)	0.58 (0.12)	3.11
LGCP	9.90 (3.66)	9.32 (2.41)	19.34 (11.45)	5.30 (1.02)	18.18 (8.65)	36.73 (4.02)	9.68 (2.67)	2.87
ICM	0.87 (0.27)	1.36 (0.35)	0.91 (0.45)	1.19 (0.40)	0.69 (0.11)	12.30 (3.02)	0.93 (0.17)	44.13

	Empirical Coverage (EC)						
	1	**2**	**3**	**4**	**5**	**6**	**7**
MCPM-N	0.99/0.80	**1.00**/0.73	0.97/**0.71**	**1.00**/0.73	0.98/0.61	**1.00**/**1.00**	0.99/**0.87**
MCPM-GP	**1.00**/**0.87**	**1.00**/**0.74**	**1.00**/**0.71**	**1.00**/**0.95**	**1.00**/**0.88**	0.80/**1.00**	**1.00**/0.85
LGCP	0.86/0.29	0.76/0.20	0.86/0.29	0.82/0.37	0.68/0.25	0.94/0.00	0.83/0.21
ICM	0.68/0.73	0.75/0.50	0.64/0.52	0.79/0.65	0.59/0.78	0.93/0.86	0.841/0.64

Table 2: *Right:* CRIME dataset. Performance on the missing regions. Time in seconds per epoch. Lower values of NLPL are better. *Left:* In-sample/Out-of-sample 90% CI coverage for the predicted event counts distributions. Higher values of EC are better.

Transfer experiment - Bovine tuberculosis (BTB) dataset

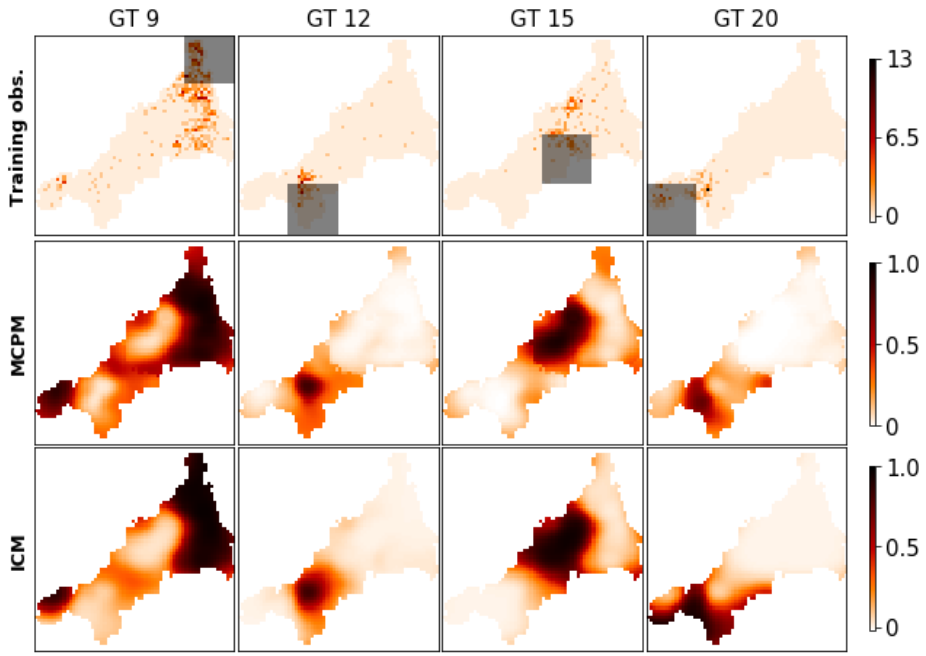


Figure 6: *First row:* Counts of the BTB incidents on a 64×64 grid. Shaded areas represent missing data regions. Estimated conditional probabilities by MCPM (*second row*) and by ICM (*third row*).

Table 3: *Upper:* RMSE and NLPL on BTB with missing data. Time in seconds per epoch. Lower values of NLPL are better. *Lower:* In-sample/Out-of-sample 90% CI coverage for the predicted event counts distributions. Higher values of EC are better.

	RMSE				NLPL (per cell)				CPU time
	gt 9	**gt 12**	**gt 15**	**gt 20**	**gt 9**	**gt 12**	**gt 15**	**gt 20**	
MCPM-N	0.83 (0.15)	0.24 (0.07)	0.28 (0.07)	0.29 (0.10)	**1.23** (0.40)	0.20 (0.07)	**0.33** (0.11)	**0.35** (0.16)	7.73
MCPM-GP	**0.81** (0.14)	0.22 (0.08)	**0.27** (0.07)	**0.27** (0.09)	1.42 (0.42)	0.27 (0.09)	0.41 (0.14)	0.58 (0.24)	**7.63**
LGCP	1.37 (0.33)	0.61 (0.13)	0.63 (0.12)	1.24 (0.56)	1.70 (0.39)	0.48 (0.11)	0.72 (0.17)	0.86 (0.36)	8.76
ICM	0.91 (0.15)	**0.21** (0.07)	0.32 (0.08)	7.24 (5.48)	1.44 (0.40)	**0.18** (0.06)	0.34 (0.10)	0.37 (0.14)	67.06

	Empirical Coverage (EC)			
	gt 9	**gt 12**	**gt 15**	**gt 20**
MCPM-N	0.87/**0.92**	0.97/**0.99**	0.93/0.96	0.95/**1.00**
MCPM-GP	**0.93**/0.91	**0.98**/0.98	**0.97**/**0.98**	**0.97**/0.99
LGCP	0.91/0.79	0.97/0.98	**0.97**/0.97	0.96/0.98
ICM	0.90/0.84	0.96/0.98	0.95/0.96	0.96/0.96

Contributions:

- Multi-task GP models capture the complexity of different social processes
- Different multi-task GP models can be used to model the intensity function of a point process
- MCPM increases modeling flexibility wrt MLGCP and ICM by introducing stochastic mixing weights
- MCPM offers a fully Bayesian model that propagates uncertainty
- Posterior inference requires approximation when the likelihood is Poisson
- Scalable inference framework ($\mathcal{O}(M^3)$) considering prior and posterior distributions for **F** and **W** separately.

Efficient Inference in Multi-task Cox Process Models

Efficient Inference in Multi-task Cox Process Models

Virginia Aglietti
University of Warwick
The Alan Turing Institute

Theodoros Damoulas
University of Warwick
The Alan Turing Institute

Edwin V. Bonilla
CSIRO's Data61
UNSW

Abstract

We generalize the log Gaussian Cox process (LGCP) framework to model multiple correlated point data jointly. The observations are treated as realizations of multiple LGCPs, whose log intensities are given by linear combinations of latent functions drawn from Gaussian process priors. The combination coefficients determine event occurrences. Among these modeling approaches, the log Gaussian Cox process (LGCP, Møller et al., 1998) is one of the most well-established frameworks, where the intensity is driven by a Gaussian process prior (GP, Williams and Rasmussen, 2006). The flexibility of LGCP comes at the cost of incredibly hard inference challenges due to its doubly-stochastic nature and the notorious scalability issues of GP models. The computational problems are

What about more difficult correlation structure?

How can we write a multi-task GP model that incorporates the correlation structure existing in a causal graph?

Multi-task Causal Learning with Gaussian Processes

Virginia Aglietti
University of Warwick
The Alan Turing Institute
V.Aglietti@warwick.ac.uk

Theodoros Damoulas
University of Warwick
The Alan Turing Institute
T.Damoulas@warwick.ac.uk

Mauricio Álvarez
University of Sheffield
Mauricio.Alvarez@sheffield.ac.uk

Javier González
Microsoft Research Cambridge
Gonzalez.Javier@microsoft.com

Abstract

This paper studies the problem of learning the correlation structure of a set of intervention functions defined on the directed acyclic graph (DAG) of a causal model. This is useful when we are interested in jointly learning the causal effects of interventions on different subsets of variables in a DAG, which is common in field

Thanks for your attention!