A Multitask Point Process Predictive Model

Wenzhao Lian[1] WL89@DUKE.EDU
Ricardo Henao[1] R.HENAO@DUKE.EDU
Vinayak Rao[2] VARAO@PURDUE.EDU
Joseph Lucas[3] JOE@STAT.DUKE.EDU
Lawrence Carin[1] LCARIN@DUKE.EDU
[1]Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, USA
[2]Department of Statistics, Purdue University, West Lafayette, IN 47907, USA
[3]Center for Predictive Medicine, Duke Clinical Research Institute, Durham, NC 27708, USA

Abstract

Point process data are commonly observed in fields like healthcare and the social sciences. Designing predictive models for such event streams is an under-explored problem, due to often scarce training data. In this work we propose a multitask point process model, leveraging information from all tasks via a hierarchical Gaussian process (GP). Nonparametric learning functions implemented by a GP, which map from past events to future rates, allow analysis of flexible arrival patterns. To facilitate efficient inference, we propose a sparse construction for this hierarchical model, and derive a variational Bayes method for learning and inference. Experimental results are shown on both synthetic data and as well as real electronic health-records data.

1. Introduction

Point process data have seen increased attention in fields like biomedical research (Rad & Paninski, 2011; Lian et al., 2014), electronic commerce (Xu et al., 2014), and healthcare analysis (Lasko, 2014). One thread of work focuses on learning arrival rates by imposing smoothness on a latent rate function (Adams et al., 2009; Rao & Teh, 2011; Lloyd et al., 2014). Another consists of predicting future arrivals as a direct function of past observations (Pillow et al., 2008; Gunawardana et al., 2011). Taking healthcare analysis as a motivating example, we focus on the latter problem: given a patient's hospital visit history up to time t, (i) when will the next visit happen? and (ii) how many visits will the patient have in $[t, t+L]$? Answering such questions provides a quantitative evaluation of the patient's risk, which helps to make treatment plans and allocate hospital resources efficiently (Amarasingham et al., 2010). Similar problems also arise in other fields, such as predicting purchasing behavior for individual customers, or predicting failures in distributed computer systems.

A few works have explored the prediction problem in point processes by learning a functional mapping from history features to the current intensity rate (Pillow et al., 2008; Rajaram et al., 2005; Gunawardana et al., 2011). In Gunawardana et al. (2011), the intensity function is constrained to be piecewise-constant, learned using decision trees and used for prediction. In our proposed multitask point process model, we build upon this piecewise-constant intensity model. To allow for flexibility of the function mapping from history features to future rate, and to capture the uncertainty of estimation, we use a nonparametric method, by imposing a Gaussian process (GP) prior on the intensity rate. However, when building such predictive models for event arrival processes, one difficulty is that the available training data are scarce for each subject/task. Therefore, we treat each individual arrival process as a task and follow a *multitask learning* approach to share information from all tasks in a *hierarchical* manner.

Methods for learning GPs from multiple tasks have been proposed (Yu et al., 2005), but involve a shared global mean function, inferred at all observed inputs (history features) across all the tasks. The posterior of this function cannot be directly updated due to non-conjugacy of the point process likelihood to GP priors. One approximation method, variational Bayes, is often applied, leading to the number of unknown parameters scaling as $\mathcal{O}(N^2)$, where N is the number of unique features from all tasks. Borrowing from the framework of *pseudo inputs* in the GP literature (Snelson & Ghahramani, 2006; Titsias, 2009), we constrain the rate functions to an M-dimensional latent space, where

Proceedings of the 32^{nd} *International Conference on Machine Learning*, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

$M \ll N$, reducing the parameter space to $\mathcal{O}(M^2 + MP)$, where P is the GP input dimension. By adjusting the locations of the pseudo inputs, we can effectively share data across tasks and efficiently represent these functions.

Learning the history-to-rate mapping functions allows one to predict future events by analytical integration or forward sampling. We consider both, evaluating our model and inference methodology on both synthetic and a real Electronic Health Records (EHR) dataset. The latter involves different categories of health problems, and we demonstrate that future hospital visits for some types of diseases are predictable even with simple history features. Our work has two main contributions: (i) providing an efficient approach to share data/parameters in hierarchical/multitask GP models; and (ii) building a predictive model for arrival data from multiple event streams, using point processes in a multitask scheme.

2. Related Work

GP-modulated point processes are a popular approach for modeling event streams (Adams et al., 2009; Rao & Teh, 2011; Lloyd et al., 2014; Lasko, 2014). Assuming a smoothly varying intensity function, the intensity rate, as well as its uncertainty, can be estimated from observed streams. Extrapolation can be used for short-term prediction. There are two main limitations of these models. First, the smoothness assumption does not hold in many scenarios. Sudden rate changes often happen upon event arrivals, *e.g.*, the risk of a patient's hospital visit might change significantly after a single visit. Second, these models involve a common modulating GP, so that multiple streams have to be aligned, something not always appropriate or possible. For example, similar arrival patterns might appear at different periods of time for different streams, which cannot be captured by such models.

Another relevant line of work on point process predictive modeling comes from the neural decoding literature (Kulkarni & Paninski, 2007; Rad & Paninski, 2011; Pillow et al., 2008), where multiple subjects/neurons are affected by common stimuli (features), generating event streams. A generalized linear model can be trained on the stimuli or spiking history to learn the rate function, and further predict future spiking events. Meanwhile, the network structure across neurons can be inferred, which in turn, helps with prediction. However, temporal alignment is also assumed in these models, which is valid in the setting of neural decoding, but not in the cases we consider.

Also related is the work of Weiss & Page (2013), which integrates the multiplicative forest Continuous Time Bayesian Network (mfCTBN) with the piecewise-constant intensity model of Gunawardana et al. (2011). More precisely, they learn forests mapping from demographic and event history features to intensity rates, but under the assumption that all subjects have the same function, *i.e.*, each event stream is a realization trajectory of some underlying model. In our work, we consider a different setting, where the variability across subjects cannot be ignored. This difference is crucial in scenarios such as healthcare analysis and electronic commerce, where variability among members of a population affects arrival pattern discovery and predictive performance.

3. Piecewise-constant Conditional Intensity Model

Multi-task point process observations are sequences of arrival time stamps $\{y_n^u\}$, with $n = 1, \cdots, D^u$ and $u = 1, \cdots, U$, where y_n^u represents the time stamp of the n-th arrival of subject/task u. Our goal is to model the sequences and make predictions for future event arrivals.

The event streams can be naturally modeled using an intensity model, with a hazard rate function $\gamma(t)$. In an infinitesimal time interval Δ, the probability of an event occuring in this interval is given by $\Delta\gamma(t)$. To build the temporal dependency between past observations and future events, we choose the hazard rate as a function of past observations $\boldsymbol{h}(t)$, denoted as $\gamma(t|\boldsymbol{h}(t))$, in which $\boldsymbol{h}(t)$ summarizes the history of past observations (Gunawardana et al., 2011); details on the potential form of $\boldsymbol{h}(t)$ are discussed subsequently. Denoting $\mathcal{Y}^u = \{y_1^u, \cdots, y_{D^u}^u\}$ as the set of arrival time stamps in task u, the likelihood is

$$p(\mathcal{Y}^u) = \prod_{n=1}^{D^u} \gamma^u(y_n^u|\boldsymbol{h}^u(y_n^u)) \exp(-\int_{y_n^u}^{y_{n+1}^u} \gamma^u(\tau|\boldsymbol{h}^u(\tau))d\tau) .$$

Assuming $\gamma^u(t|\boldsymbol{h}^u(t))$ as piecewise constant with N^u change points (pieces) at $\{t_i^u\}_{i=1}^{N^u}$, and piece length $\Delta_i^u = t_{i+1}^u - t_i^u$, we have the likelihood

$$\begin{aligned} p(\mathcal{Y}^u) = & \prod_{i=1}^{N^u} \gamma^u(t_i^u|\boldsymbol{h}^u(t_i^u))^{\mathbb{I}(t_i^u \in \mathcal{Y}^u)} \\ & \times \exp\{-\Delta_i^u \gamma^u(t_i^u|\boldsymbol{h}^u(t_i^u))\} . \end{aligned} \tag{1}$$

Many approaches exist to extract features $\boldsymbol{h}(t)$ from past event arrivals. One possible feature-construction approach uses empirical rates at recent time points (Rajaram et al., 2005), *i.e.*, $\boldsymbol{h}^u(t_i^u) \in \mathbb{R}_+^P$ with $h_p^u(t_i^u) = \bar{\gamma}(t_i^u - L_p)$ for $L_1, L_2, \cdots, L_P$ predefined lengths of memory. Here $\bar{\gamma}(t_i^u - L_p)$ refers to the empirically computed rate at time $t_i^u - L_p$. Keeping the algorithm simple, we adopt a construction similar in spirit to Gunawardana et al. (2011): $\boldsymbol{h}^u(t_i^u)$ is P dimensional, where its p-th element denotes the count of arrivals in $[t_i^u - L_p, t_i^u]$. Therefore, $\boldsymbol{h}^u(t)$

is a piecewise constant function. Revisiting the likelihood in (1), we observe that N^u is the number of change points (pieces) of the feature function $\boldsymbol{h}^u(t)$ going through the whole event stream $\{y_n^u\}_{n=1}^{D^u}$, and that N^u scales linearly with D^u (the number of events). Side information in the form of covariates can also be considered as a natural extension, by augmenting the feature space. Constructing richer features from history observations is left as an interesting direction for future work.

To complete the intensity model, the functional mapping from feature space to intensity rate must be specified. Define the space of the history features $\boldsymbol{h}^u(t_i^u)$ as $\mathcal{H}$, where each point $\boldsymbol{h}^u(t_i^u) \triangleq \boldsymbol{h}_i^u \in \mathcal{H}$ is a possible feature vector. Here we impose a GP prior on a set of functions $f^u(\cdot) : \mathcal{H} \to \mathbb{R}$, followed by a transformation to ensure the non-negativity of intensity rates. We use a square transformation ($\gamma^u(\cdot) = \{f^u(\cdot)\}^2$) rather than the common $\gamma^u(\cdot) = \exp\{f^u(\cdot)\}$, because the uncertainty of $f^u(\cdot)$ cannot be properly estimated in the latter construction, as discussed by Lloyd et al. (2014); this issue is addressed thoroughly in Section 5. To resolve the ambiguity caused by $\{f^u(\cdot)\}^2 = \{-f^u(\cdot)\}^2$, a prior specification favoring the positive half-space is imposed and works well in practice.

4. Multitask Point Process Predictive Model

As mentioned in Section 1, sharing information across tasks is necessary when learning the rate functions from sparse data. Accordingly, we consider a hierarchical GP construction. Denote the rate at feature vector $\boldsymbol{h}_i^u$ as γ_i^u, with the corresponding transformed rate denoted as $f_{N,i}^u$. The generative process from history features to rates may be described as follows

$$\begin{aligned} \boldsymbol{\mu}_N^0 &\sim \mathcal{GP}\left(\boldsymbol{g}, \frac{1}{\xi}\boldsymbol{K}_{NN}\right), & (2)\\ \boldsymbol{f}_N^u &\sim \mathcal{N}\left(\boldsymbol{\mu}_N^0, \boldsymbol{K}_{NN}\right), & (3)\\ \gamma_i^u &= \{f_{N,i}^u\}^2 . & (4) \end{aligned}$$

In (2), $\boldsymbol{g}$ is a prior mean, which may be set according to prior knowledge or the empirical average intensity rate over all tasks. Parameter ξ controls the complexity of the hyperprior. The covariance matrix $\boldsymbol{K}_{NN}$ can be obtained from a squared-exponential or Matérn function with automatic relevance determination (ARD) covariance kernel, for example,

$$\boldsymbol{K}_{NN,ij} = \tau^2 \exp\left(-\sum_{p=1}^{P} \frac{(h_{pi} - h_{pj})^2}{2\lambda_p^2}\right) + \sigma^2 \mathbb{I}(i = j), \quad (5)$$

where $\boldsymbol{h}_i \in \mathcal{H}$ may be any possible history feature. In (2), $\boldsymbol{\mu}_N^0$ determines the global mean function of transformed rate functions, $\boldsymbol{f}_N^u$, from all U tasks. For each individual task, the task-specific transformed rate function, $\boldsymbol{f}_N^u$, is a "noisy version" of the global mean function $\boldsymbol{\mu}_N^0$. Our construction is inspired by Yu et al. (2005), which can be shown analogous to a hierarchical construction of the multitask linear regression model

$$\begin{aligned} \boldsymbol{w}^0 &\sim \mathcal{N}\left(\boldsymbol{v}, \frac{1}{\xi}\boldsymbol{I}\right), & (6)\\ \boldsymbol{w}^u &\sim \mathcal{N}\left(\boldsymbol{w}^0, \boldsymbol{I}\right), & (7)\\ f_{N,i}^u &\sim \mathcal{N}\left({\boldsymbol{w}^u}^\top \boldsymbol{h}_i, \sigma^2\right), & (8) \end{aligned}$$

where $\boldsymbol{w}^0$ and $\boldsymbol{w}^u$ are the global and task-specific regression parameters, respectively, and $\boldsymbol{v}$ is the hypermean for $\boldsymbol{w}^0$. Choosing a linear kernel for (2) and (3) ($\boldsymbol{K}_{NN,ij} = \boldsymbol{h}_i^T \boldsymbol{h}_j$), and letting $\boldsymbol{v} = \boldsymbol{g} = 0$, the construction through (2)-(3) is equivalent to the multitask linear regression model in (6)-(8) on a finite observed dataset.

If there exists enough training data, *i.e.*, $\{\boldsymbol{h}_i^u\}_{i=1}^{N^u}$ densely covers the feature space $\mathcal{H}$, $\boldsymbol{f}_N^u$ is mainly determined by the data in task u. Otherwise, $\boldsymbol{f}_N^u$ is significantly affected by the prior $\boldsymbol{\mu}_N^0$, which pools together information from rate functions $\{\boldsymbol{f}_N^u\}$ from all tasks.

However, a problem with the hierarchal GP construction using (2) and (3) is that the dimension of $\boldsymbol{\mu}_N^0$ and $\boldsymbol{f}_N^u$ is typically massive, in fact it is not smaller than the number of unique history features from all tasks, *i.e.*, $|\cup_u \{\boldsymbol{h}_i^u\}_{i=1}^{N^u}|$, where $|\mathcal{S}|$ refers to the cardinality of a set $\mathcal{S}$. Thus, the number of unknown parameters in the model scales with $\mathcal{O}(U \sum_{u=1}^{U} N^u)$, or $\mathcal{O}(U(\sum_{u=1}^{U} N^u)^2)$ if variance estimation is also considered. In point process models in particular, this is impractical because the influence of the likelihood is weak and noise levels are high. Therefore, we propose an alternative approach inspired by the pseudo input framework for GPs (Snelson & Ghahramani, 2006; Titsias, 2009). Specifically, we propose a two-step process by introducing an M-dimension vector $\boldsymbol{f}_M^u$ for each task, where $M \ll |\cup_u \{\boldsymbol{h}_i^u\}_{i=1}^{N^u}|$. Then, using a so-called *conditional Gaussian Processes*, we define the generative process as

$$\begin{aligned} \boldsymbol{\mu}_M^0 &\sim \mathcal{N}\left(\boldsymbol{g}, \frac{1}{\xi}\boldsymbol{K}_{MM}\right), & (9)\\ \boldsymbol{f}_M^u &\sim \mathcal{N}\left(\boldsymbol{\mu}_M^0, \boldsymbol{K}_{MM}\right), & (10)\\ \boldsymbol{f}_N^u | \boldsymbol{f}_M^u &\sim \mathcal{GP}\Big(\boldsymbol{g} + \boldsymbol{K}_{NM}^u \boldsymbol{K}_{MM}^{-1}(\boldsymbol{f}_M^u - \boldsymbol{g}), & (11)\\ &\qquad (1 + \frac{1}{\xi})(\boldsymbol{K}_{NN}^u - \boldsymbol{K}_{NM}^u \boldsymbol{K}_{MM}^{-1} {\boldsymbol{K}_{NM}^u}^\top)\Big), \end{aligned}$$

where $\boldsymbol{f}_N^u$ contains the transformed rates at all possible feature vectors $\{\boldsymbol{h}_i^u\}_{i=1}^{N^u}$ in task u, while $\boldsymbol{f}_M^u$ is a M-dimensional vector, consisting of the transformed rates at M feature-vector locations $\{\boldsymbol{s}_i\}_{i=1}^{M}$.

In (11), for each task u, only the function values at feature vectors appearing in task u are required, while in (3), for

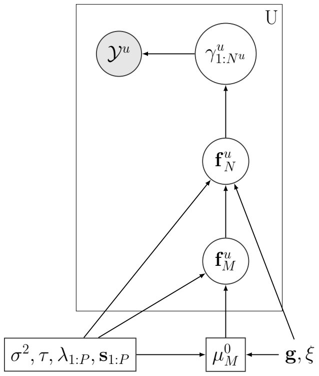

Figure 1. Graphical model representation (filled circle are observed, empty circles denotes latent variables, rectangles refer to model parameters, and the rest are free hyper-parameters).

each task, the function values at all feature vectors appearing in all tasks need to be specified.

Following the GP literature (Titsias, 2009), we refer to $\{\boldsymbol{s}_i\}$ as pseudo inputs. These need not appear in any of the tasks (or even $\mathcal{H}$); we only require that a distance can be properly defined between $\boldsymbol{s}_i$ and $\boldsymbol{h}^u_j$. And similar to (5), $\boldsymbol{K}_{MM}$ and $\boldsymbol{K}^u_{NM}$ can be obtained through the squared-exponential or the Matérn function with ARD kernel, with $\boldsymbol{h}^u_j$ and $\boldsymbol{s}_i$ as covariates. Specifically,

$$\boldsymbol{K}_{MM,ij} = \tau^2 \exp\left(-\sum_{p=1}^{P} \frac{(s_{pi} - s_{pj})^2}{2\lambda_p^2}\right) + \sigma^2 \mathbb{I}(i = j)\,,$$

$$\boldsymbol{K}^u_{NM,ij} = \tau^2 \exp\left(-\sum_{p=1}^{P} \frac{(h^u_{pi} - s_{pj})^2}{2\lambda_p^2}\right)\,.$$

It can be shown that the construction through (9)-(11) results in the same marginal prior distribution for $\boldsymbol{f}^u_N$ as (2)-(3) (see supplemental material) while having a significantly reduced computational cost. Overfitting problems can also be alleviated because model complexity is reduced. This is especially beneficial when sophisticated features are constructed, resulting in very large history feature spaces, however, with the rate function likely living in a low-dimensional manifold. As a result, assuming $\{\boldsymbol{s}_i, f^u_{M,i}\}_{i=1}^{M}$ captures the characteristics of the function $\{\boldsymbol{h}^u_i, f^u_{N,i}\}_{i=1}^{N^u}$, while achieving computational savings, and improved estimation accuracy. The proposed full generative model is summarized by equations (9)-(11), (4) and (1), as demonstrated in Figure 1.

5. Inference

Model parameters $\boldsymbol{\Theta}$ include the pseudo-input locations, the global mean of transformed rates, and the GP hyper-paramters: $\boldsymbol{\Theta} = \{\{\boldsymbol{s}_m\}_{m=1}^{M}, \boldsymbol{\mu}^0_M, \sigma^2, \tau^2, \{\lambda_p\}_{p=1}^{P}\}$. Obtaining a full posterior distribution for $\boldsymbol{\mu}^0_M$ is straightforward due to local conjugacy. However we observed no significant difference in performance against just a point estimate, and for easier interpretability, focus on the latter.

Maintaining $p(\boldsymbol{f}^u_M, \boldsymbol{f}^u_N | \mathcal{Y}, \boldsymbol{\Theta})$, the full posterior distribution over the intensity functions of each task is important for transfer learning across tasks with different numbers of observations. However, this is not straightforward due to the non-conjugate point process likelihood, and borrowing ideas from variational learning for sparse GPs (Lloyd et al., 2014; Titsias, 2009), we propose a variational form to approximate the posterior $p(\boldsymbol{f}^u_M, \boldsymbol{f}^u_N | \mathcal{Y}, \boldsymbol{\Theta})$. Letting $\mathcal{Y}$ refer to the complete collection of event streams $\mathcal{Y}^1, \cdots, \mathcal{Y}^U$, we use

$$\begin{aligned} q(\boldsymbol{f}^u_N, \boldsymbol{f}^u_M) &= p(\boldsymbol{f}^u_N | \boldsymbol{f}^u_M) q(\boldsymbol{f}^u_M) \\ &= p(\boldsymbol{f}^u_N | \boldsymbol{f}^u_M) \mathcal{N}(\boldsymbol{f}^u_M; \boldsymbol{\mu}^u, \boldsymbol{\Sigma}^u)\,. \end{aligned} \tag{12}$$

Note that we allow a free-form Gaussian distribution for $\boldsymbol{f}^u_M$, the task-specific transformed rates at the pseudo inputs. However the transformed rates evaluated at all features in each task, $\boldsymbol{f}^u_N$, are constrained by the low-dimensional function $\boldsymbol{f}^u_M$ via (11). In the following, we use a simplified notation q^u to denote $q(\boldsymbol{f}^u_M) = \mathcal{N}(\boldsymbol{f}^u_M; \boldsymbol{\mu}^u, \boldsymbol{\Sigma}^u)$. Because of the special factorized form in (12), $\{\boldsymbol{\mu}^u, \boldsymbol{\Sigma}^u\}$ are the only variational parameters in the algorithm.

The inference objective is to maximize the variational lower bound (Beal, 2003) (called ELBO for evidence lower bound optimization):

$$\begin{aligned} \log p(\mathcal{Y}, \boldsymbol{\Theta}) \geq\ & \log p(\boldsymbol{\Theta}) \\ & + \sum_{u=1}^{U} \left\{ \mathbb{E}_{q(\boldsymbol{f}^u_N, \boldsymbol{f}^u_M)} [\log \frac{p(\mathcal{Y}^u, \boldsymbol{f}^u_N, \boldsymbol{f}^u_M)}{q(\boldsymbol{f}^u_N, \boldsymbol{f}^u_M)}] \right\}\,. \end{aligned} \tag{13}$$

Note that in (13), $q(\boldsymbol{f}^u_N, \boldsymbol{f}^u_M)$ and $p(\mathcal{Y}^u, \boldsymbol{f}^u_N, \boldsymbol{f}^u_M)$ implicitly depend on $\boldsymbol{\Theta}$. Since we only impose a Gaussian prior on $\boldsymbol{\mu}^0_M$ in (9), $\log p(\boldsymbol{\Theta})$ is simplified to $\log p(\boldsymbol{\mu}^0_M)$. Priors on other model parameters may also be imposed, *e.g.*, a log-normal prior on GP hyper-parameters, but here we learn their maximum likelihood estimate (MLE) instead. To show explicit dependence, we denote the ELBO as $\mathcal{F}(q^1, \cdots, q^U, \boldsymbol{\Theta})$. We maximize $\mathcal{F}(q^1, \cdots, q^U, \boldsymbol{\Theta})$ in (13), giving a variational Expectation Maximization (EM) algorithm guaranteed to converge to a local optimum (Beal, 2003). In practice, we alternate between a variational E-step, where $\boldsymbol{\Theta}$ is fixed and $\mathcal{F}(q^1, \cdots, q^U, \boldsymbol{\Theta})$ is maximized *w.r.t.* $\{q^u\}_{u=1}^{U}$, and a variational M-step, where $\{q^u\}_{u=1}^{U}$ is fixed and $\mathcal{F}(q^1, \cdots, q^U, \boldsymbol{\Theta})$ is maximized *w.r.t.* $\boldsymbol{\Theta}$. Derivation details are standard; we list the key steps below:

$$\mathcal{F}(q^1, \cdots, q^U, \boldsymbol{\Theta}) = \sum_{u=1}^{U} \mathbb{E}_{q^u} \left[\mathbb{E}_{p(\boldsymbol{f}_N^u | \boldsymbol{f}_M^u)} [\log p(\mathcal{Y}^u | \boldsymbol{f}_N^u)] \right] + \sum_{u=1}^{U} \mathbb{E}_{q^u} \left[\log \frac{p(\boldsymbol{f}_M^u | \boldsymbol{\mu}_M^0)}{q(\boldsymbol{f}_M^u | \boldsymbol{\mu}^u, \boldsymbol{\Sigma}^u)} \right] + \log p(\boldsymbol{\mu}_M^0) \triangleq \mathcal{F}_1 + \mathcal{F}_2 + \mathcal{F}_3 \,. \quad (14)$$

The first term $\mathcal{F}_1$ (15) measures how the functions specified by q^u (determining the distribution of function values at pseudo inputs) fit the observations. Because of the sparsity assumption of the conditional Gaussian process, we can integrate out $\boldsymbol{f}_N^u$ and leave $\mathcal{F}_1$ as a function of only $\{\boldsymbol{\mu}^u, \boldsymbol{\Sigma}^u\}$ and $\boldsymbol{\Theta}$:

$$\mathcal{F}_1 = \sum_{u=1}^{U} \int q(\boldsymbol{f}_N^u | \boldsymbol{\mu}^u, \boldsymbol{\Sigma}^u) \log p(\mathcal{Y}^u | \boldsymbol{f}_N^u) d\boldsymbol{f}_N^u \quad (15)$$
$$= \sum_{u=1}^{U} \sum_{i=1}^{N^u} \{ \mathbb{I}(t_i^u \in \mathcal{Y}^u) \mathbb{E}[\log(f_{N,i}^u)^2] - \Delta_i^u \mathbb{E}[(f_{N,i}^u)^2] \} \,,$$

The expectation is *w.r.t.* $q(\boldsymbol{f}_N^u | \boldsymbol{\mu}^u, \boldsymbol{\Sigma}^u) = \mathcal{N}(\boldsymbol{b}^u, \boldsymbol{B}^u)$, with parameters in (16) and (17):

$$\boldsymbol{b}^u = \boldsymbol{g} + \boldsymbol{K}_{NM}^u \boldsymbol{K}_{MM}^{-1} (\boldsymbol{\mu}^u - \boldsymbol{g}) \,, \quad (16)$$
$$\boldsymbol{B}^u = \left(1 + \frac{1}{\xi}\right) \left(\boldsymbol{K}_{NN}^u - \boldsymbol{K}_{NM}^u \boldsymbol{K}_{MM}^{-1} \boldsymbol{K}_{NM}^{u\;\top} \right) + \boldsymbol{K}_{NM}^u \boldsymbol{K}_{MM}^{-1} \boldsymbol{\Sigma}^u (\boldsymbol{K}_{NM}^u \boldsymbol{K}_{MM}^{-1})^\top \,. \quad (17)$$

$\mathbb{E}[\log(f_{N,i}^u)^2]$ in (15) can be calculated using confluent hypergeometric functions (Lloyd et al., 2014). Further details can be found in the Supplemental Material, where we also provide a robust approximation to tackle the well-known numerical instability issue in confluent hypergeometric function evaluations (Ancarani & Gasaneo, 2008).

As a side note on why we prefer a squared transformation over the exponential, when using the latter, $\mathcal{F}_1$ is modified as (18), denoted as $\tilde{\mathcal{F}}_1$:

$$\tilde{\mathcal{F}}_1 = \sum_{u=1}^{U} \sum_{i=1}^{N^u} \{ \mathbb{I}(t_i^u \in \mathcal{Y}^u) \mathbb{E}[f_{N,i}^u] - \Delta_i^u \mathbb{E}[\exp(f_{N,i}^u)] \} \,, \quad (18)$$

Now the variance of $f_{N,i}^u$ does not affect $\mathbb{E}[f_{N,i}^u]$ at event times $t_i^u \in \mathcal{Y}^u$ (Lloyd et al., 2014), but only contributes to the second term through $\mathbb{E}[\exp(f_{N,i}^u)]$. This leads to instability issues during inference. We also implemented the algorithm using this transformation, but the estimated function f^u diverged even with strong prior constraints.

$\mathcal{F}_2$, as specified in (19), penalizes the task-specific function's deviation from the global mean function $\boldsymbol{\mu}_M^0$. This is especially important for tasks with few training data, *e.g.*, short event streams or few event arrivals.

$$\mathcal{F}_2 = -\tfrac{U}{2} \log |\boldsymbol{K}_{MM}| + \tfrac{1}{2} \sum_{u=1}^{U} \log |\boldsymbol{\Sigma}^u| \quad (19)$$
$$- \textstyle\sum_{u=1}^{U} \tfrac{1}{2} tr \left[\boldsymbol{K}_{MM}^{-1} \left(\boldsymbol{\mu}^u \boldsymbol{\mu}^{u\top} + \boldsymbol{\Sigma}^u + \boldsymbol{\mu}_M^0 \boldsymbol{\mu}_M^{0\;\top} - 2 \boldsymbol{\mu}^u \boldsymbol{\mu}_M^{0\;\top} \right) \right] .$$

$\mathcal{F}_3$ involves the hyper-prior on the global function $\boldsymbol{\mu}_M^0$, with ξ controlling the belief strength:

$$\mathcal{F}_3 = \frac{1}{2} \{ \log |\xi \boldsymbol{K}_{MM}^{-1}| - tr(\xi \boldsymbol{K}_{MM}^{-1} (\boldsymbol{\mu}_M^0 - \boldsymbol{g})(\boldsymbol{\mu}_M^0 - \boldsymbol{g})^\top) \} \,.$$

Having defined the variational objective, we can maximize the objective alternately *w.r.t.* to $\{\boldsymbol{\mu}^u, \boldsymbol{\Sigma}^u\}$ and $\boldsymbol{\Theta}$. In the variational E-step, we update the variational parameters $\boldsymbol{\mu}^u$ and $\boldsymbol{\Sigma}^u$ using gradient descent methods. As a practical consideration, to preserve the positive definiteness of $\boldsymbol{\Sigma}^u$, we optimize it instead over its lower Cholesky decomposition, $\boldsymbol{L}^u$, where $\boldsymbol{\Sigma}^u = \boldsymbol{L}^u \boldsymbol{L}^{u\top}$, and with positiveness constraints on the diagonal elements. In the variational M-step, updates for $\boldsymbol{\mu}_M^0$ are obtained in closed form due to local conjugacy. Gradient methods are needed for updating other parameters, including locations of pseudo inputs and GP hyperparameters (see Supplemental Material for details).

When the number of tasks is large, the algorithm can be readily distributed over a cluster, with the E-step optimization procedures for tasks distributed over individual cores/machines.

6. Experiments

We evaluate our model on both synthetic data as well as an EHR dataset. At any time t, the basic challenge is to predict the event arrival patterns in the interval $[t, t+L]$. One baseline is a simplified model where each task is learned independently (fixing $\boldsymbol{\mu}_M^0$ in (10) to $\boldsymbol{g}$) to demonstrate the benefit of sharing across tasks. We refer to our proposed model as MTPP (for multitask point process model), and the simplified model as IPP (for independent point process model). Another baseline is Poisson regression (PoiR), specialized for the prediction problem considered. Here, at each time stamp t, using observations in a window of length L_P (the maximum memory length), we calculate the history features $\boldsymbol{h}(t)$, and treat them as predictors for the number of arrivals in the succeeding window $[t, t+L]$. While this is not a generative processs, the Poisson regression model can be trained using observations and tested on prediction tasks.

6.1. Synthetic Experiments

For the first experiment, we synthesize a rate function for each of $U = 10$ tasks using a two step process. First the rates at $M = 10$ feature locations (pseudo inputs) are generated from a common Gaussian distribution. The rate

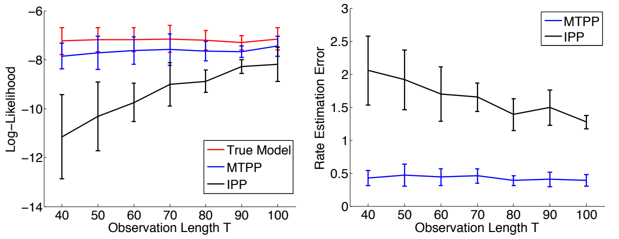

Figure 2. Left: Predictive log-likelihood per unit time; Right: mean normalized estimation error of intensity rates.

function is then drawn conditioned on these, as specified by (11). The features constructed at time t are the event counts in intervals $[t-5, t]$ and $[t-1, t]$. The GP hyperparameters are set as, $\lambda_1 = \lambda_2 = 4, \tau = 1.2$, and $\sigma = 0.01$, to produce moderate rate functions, *e.g.*, the empirical average rate over each event stream is between 0.05 and 1.5. Using the conditional intensity model introduced in Section 3, we generate event streams for each of the U tasks. We train the models using observations up to time T, and perform forward prediction.

For inference, the pseudo points were initialized using K-means clustering over the history features appearing in the training set. The GP hyperparameters are initialized setting the length-scale parameters, $\{\lambda_p\}$, as the standard deviation of observed feature vectors, the magnitude, τ, as the standard deviation of square roots of empirical rates (computed via binning methods), and the noise term, σ^2, as a small value, $0.01\tau^2$. The algorithm converges within 20 iterations, after which the relative successive increase of the variational objective is negligible.

Model fit: We first evaluate the model and inference by comparing MTPP and IPP with the ground truth model. We vary the observation length of the training set T; the experiments for each setting are repeated for 15 runs, with mean and standard deviation reported.

The left panel of Figure 2 compares the predictive log-likelihood per unit time (data log-likelihood discounted by event stream length) on unseen streams. We see that with an increasing length of training observations, the predictive log-likelihood of both MTPP and IPP gradually approaches the log-likelihood obtained with the true model. When T is small, which is true in many practical scenarios, IPP cannot fit the data well while MTPP learns the model accurately. This is again illustrated in the right panel of Figure 2, where for both MTPP and IPP, we compute the rate estimation error (ℓ_2-norm), normalized *w.r.t.* the true rates, and averaged over all history features occurred in the data. We observe that MTPP outperforms IPP, especially in scarce-data scenarios, demonstrating the benefit of sharing across tasks.

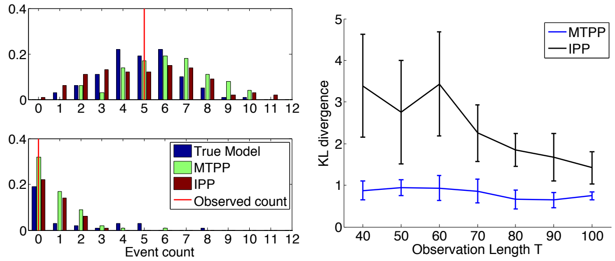

Figure 3. Left: Predicted empirical distributions of event counts in $[t, t+5]$; Right: KL divergence between predicted distributions of event counts in $[t, t+5]$ using the learned and the true model.

Forward prediction: We evaluate the prediction performance using two metrics (mirroring the two questions posed in Section 1). The first is a binary classification problem on whether at least one event occurs in $[t, t+L]$, and the second, estimating the distribution of event arrival counts in $[t, t+L]$. For the former, at time t, the probability of no event occurring until $t+L$ can be analytically computed as $\exp(-\int_t^{t+L} \gamma(\tau|\boldsymbol{h}(\tau))d\tau)$. This is easily solvable because $\boldsymbol{h}(\tau)$ is piecewise-constant. For the latter, because the intensity model is trained given observations, *i.e.*, the history-to-rate functions are learned, at time t, we can generate sample paths in a forward manner. Using Monte Carlo sampling, we estimate the quantities of interest, *e.g.*, the distribution of event arrival counts in $[t, t+L]$.

To better illustrate how the model performs forward prediction, the left panel of Figure 3, shows for one testing instance, the empirical distribution of event arrival counts obtained from 100 Monte Carlo sample paths. Qualitatively, the distribution predicted by MTPP matches the one predicted by the true model best. The actual observed count can be considered as a draw from the distribution produced by the true model. To quantitively measure the performance, we compute the KL divergence between predicted distributions using the learned models and the true model, varying with observation length T. As indicated in the right panel of Figure 3, the KL divergence between predicted distributions using IPP and the true model is large for short streams, decreasing only with longer streams. For MTPP, there is a much smaller divergence (which decreases to a value around 0.75).

We next compare prediction results from MTPP, IPP, and PoiR, where for PoiR all tasks are trained independently. The testing instances are constructed by taking a sequence of snapshots on the unseen streams. In particular, we start at T with a length L sliding window, extract history features, record the event count, and move forward with stepsize $\frac{L}{2}$ until the end of the stream. Figure 4 demonstrates the prediction performance as a function of prediction window L. Shown in the left panel is the Area Under the Curve (AUC) for the binary problem of whether at least one event occurs in $[t, t+L]$. MTPP performs best, and is closest to the AUC achieved using the true model. The in-

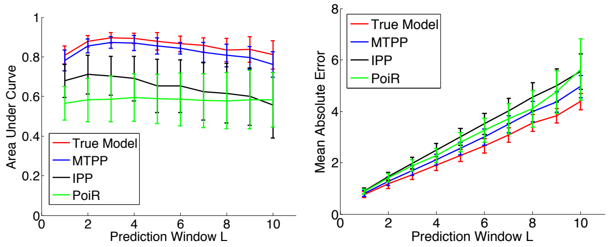

Figure 4. Left: AUC of binary prediction on whether at least one event occurs in $[t, t+L]$; Right: MAE of event arrival counts in $[t, t+L]$.

dependent models perform poorly due to insufficient training data. One interesting phenomenon is that the prediction accuracy increases first and starts decreasing after a change point. This phenomenon results from a trade-off between two effects. First, for small L, the probability of no event occurring decreases from a moderate value (around 0.5) to 0, leading to an easier prediction problem. Second, as L increases further, the accumulating Poisson noise makes it harder. When L is small, the first effect dominates leading to a higher AUC. However, after a change point, the accumulated Poisson noise dominates, driving prediction accuracy down. The right panel of Figure 4 shows the Mean Absolute Error (MAE) of event arrival counts in $[t, t+L]$. Unlike the AUC for binary prediction, the MAE for all methods increases monotonically as prediction window length increases. MTPP outperforms other baselines, achieving the MAE closest to the one obtained by the true model. The increment is linear because of the linearly accumulating Poisson noise.

6.2. Applications on Electrical Health Records

We now consider a real-world application involving an EHR dataset. We use the New Zealand national minimum dataset [1], covering the years 2007 through 2011 (inclusive). The data contains approximately 3.3 million inpatient visits from 1.5 million unique individuals with ages from 18 to 65. Available variables include ICD-10-AM (Australian Modification) diagnosis and procedure codes which are grouped into 22 broad categories (World Health Organization, 2010).

We focus on hospital visits for each disease category, associated with a block of ICD-10-AM billing codes (*e.g.* billing codes in the range C00 to D48 all refer to neoplasms). Treating clinical data as point process observations is a relatively novel approach to the best of our knowledge, and has only been adopted in Lasko (2014). In our experiments, for each disease category, we record all patients' visits associated with billing codes belonging to it, filter out patients with infrequent visits (fewer than 50), and split visit sequences for each patient into training and testing (split at the time stamp when half of the number of visits are observed).

After preprocessing, we have visit streams of multiple patients for each disease category, with the number of patients varying from 36 to 118, and the number of visits per patient varying from 51 to 98. In total, we analyze 6 disease types, listed in Table 1, mostly related with chronic diseases. For example, neoplasms include malignant and benign ones, and metabolic problems include type-I and type-II diabetes. Note we are not exploring the correlation across visits for different disease types, which is also interesting, and left as future work. The aim of this application to EHR data is to show that using the multitask point process model proposed, with a simple feature construction approach, the visit patterns for some disease types are reasonably predictable. For each patient's visit stream of a disease category, the features constructed include the number of his/her hospital visits for this disease in the previous week, month, and six months. For inference, we set the number of pseudo inputs $M = 15$, and the initialization procedure is the same as in the synthetic experiments. The algorithm generally converges in tens of iterations.

Similar with the above synthetic experiments, we evaluate the algorithm from two perspectives: model fit and predictive ability. For model fit, we evaluate the data log-likelihood, comparing with two state-of-the-art approaches using GP modulated renewal processes: a direct inference method in Lasko (2014) and a thinning approach in Rao & Teh (2011). For the prediction tasks, unlike the synthetic experiments, we do not have access to the ground truth model, *i.e.*, knowledge about heterogeneity across population/tasks. Thus, we use the test dataset to compare MTPP with PoiR, using the same model for all patients for the latter. We also evaluated IPP, but because the patients' visits are sparse, the individual models learned with this scarce training data perform poorly. Hence only results of pooled MTPP and PoiR are reported.

Model Fit: GP modulated renewal processes are trained using both direct inference as in Lasko (2014) and thinning approach as in Rao & Teh (2011), for each patient's visit sequence in the neoplasm category. We then compare these two methods with MTPP on the data log-likelihood per time unit (day) over all sequences. As shown in the top-left panel of Figure 5, MTPP is comparable with the other two methods, with a smaller variation due to the sharing across patients. Shown in the top-right and bottom panels are the mean intensity functions inferred of anonymous patients (error bars are omitted for clarity), each associated with the raster plot of the hospital visits in the bottom. Relative dates, instead of calendar dates are used for confidentiality. As can be seen, the two temporal GP methods infer

[1] http://www.health.govt.nz/nz-health-statistics

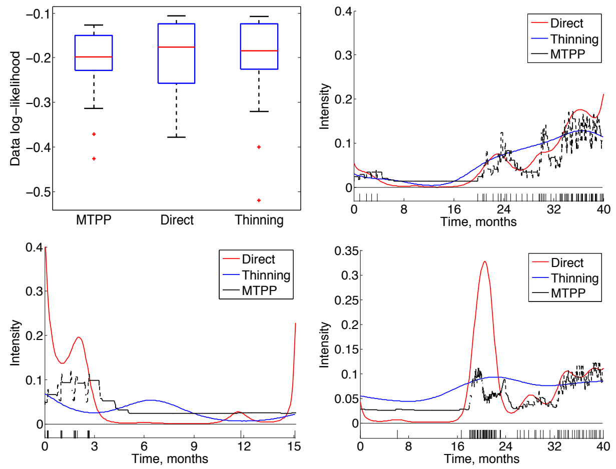

Figure 5. Model fit results. Top-left: Comparison of data log-likelihood per day of MTPP, direct inference, and thinning approach; Top-right and bottom row: Intensity functions inferred of anonymous patients' arrival sequences.

a smooth function modeling the rate, which fails to capture sudden changes. For example, see the small-to-large change rate around the 33rd month in the top-right panel. Also note in the bottom-left panel, MTPP performs well at the beginning and ending of the sequence, with no boundary effects of GPs. Such differences are consistent over the dataset. Note that the intensity function inferred via our method is piecewise-constant, with change points located where recent history changes. The goodness of fit suggests this current rate's dependence on history enables forward prediction.

Forward Prediction: Figure 6 (left panel) shows the learned history-to-rate mapping function globally shared by all patients in the neoplasm category. In particular, we plot the value of the function in terms of two features, namely, number of arrivals in windows $[t-7, t]$ and $[t-30, t]$ (the previous week and month). As indicated, a patient with many visits in a month and few visits in a week has a larger rate of a revisit.

The right panel in Figure 6 shows that for neoplasms, the MAE increases monotonically as the prediction window length increases up to 90 days. This is consistent with the results from synthetic data shown in Figure 4. Interestingly, because PoiR requires a separate model for each prediction interval, no noise variance accumulates in the model, and the increase in MAE results from the data itself. On the other hand, we just train MTPP once and generate sample paths for different window lengths, resulting in a Poisson noise that increases linearly with prediction interval. As a result, for MTPP the MAE diverges faster than for PoiR as L increases.

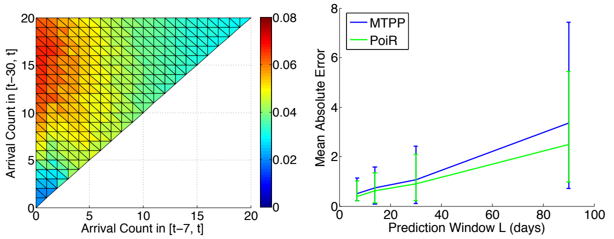

Figure 6. Neoplasms results. Left: Global rate function inferred by MTPP as a function of the number of arrivals in windows $[t-7, t]$ and $[t-30, t]$; Right: MAE as a function of the prediction window length, L, in days.

Results in Table 1 show AUC values for the 6 disease categories considered. Two settings for the length of the prediction windows, namely, 1 week and 1 month are evaluated. We see that consistent with the results for artificial data in Figure 4, AUCs tend to increase for moderate sizes of L. We also verified (results not shown) that further increasing the prediction window size has an accordingly negative impact on the AUCs. In fact, predictions become no better than random as L approaches 6 months for some disease types. When comparing MTPP to PoiR, we observe that the former outperforms the latter in 4 of 6 disease types. This supports the hypothesis that the changes of hospital visit patterns among the population are disease specific. However, correlations across diseases and clustering of patient subpopulations could be exploited, and are left as future work.

Table 1. AUC of binary predictions of events occurring in weekly and monthly windows for 6 disease types.

	1 WEEK		1 MONTH	
DISEASE TYPE	MTPP	POIR	MTPP	POIR
NEOPLASMS	**0.7379**	0.7249	**0.8136**	0.8058
METABOLIC	**0.6807**	0.6170	**0.6778**	0.6195
NERVOUS	0.6926	**0.7241**	**0.7978**	0.7167
CIRCULATORY	**0.6807**	0.6778	0.6170	**0.6195**
RESPIRATORY	0.5733	**0.6302**	0.6308	**0.6322**
DIGESTIVE	**0.6050**	0.5562	**0.6555**	0.6170

7. Conclusions and Future Work

We have considered the problem of analyzing multiple streaming point processes in a multi-task setting, and have proposed a simple predictive strategy. The proposed model, using hierarchical GPs, leverages information across the tasks in a nonparametric manner. Exploring multi-task marked point processes, designing richer feature construction approaches for predictive models, and clustering tasks based on their arrival patterns, are left as interesting open problems.

Acknowledgements

The research reported here was supported in part by ARO, DARPA, DOE, NGA and ONR.

References

Adams, Ryan P., Murray, Iain, and MacKay, David J. C. Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In *Proceedings of the 26th International Conference on Machine Learning*, 2009.

Amarasingham, Ruben, Moore, Billy J., Tabak, Ying P., Drazner, Mark H, Clark, Christopher A., Zhang, Song, Reed, W. Gary, Swanson, Timothy S., Ma, Ying, and Halm, Ethan A. An automated model to identify heart failure patients at risk for 30-day readmission or death using electronic medical record data. *Medical Care*, 48 (11):981–988, 2010.

Ancarani, L. U. and Gasaneo, G. Derivatives of any order of the confluent hypergeometric function. *Journal of Mathematical Physics*, 49, 2008.

Beal, Matthew James. *Variational algorithms for approximate Bayesian inference*. PhD thesis, U. London, 2003.

Gunawardana, Asela, Meek, Christopher, and Xu, Puyang. A model for temporal dependencies in event streams. In *Advances in Neural Information Processing Systems*, pp. 1962–1970, 2011.

Kulkarni, Jayant E. and Paninski, Liam. Common-input models for multiple neural spike-train data. *Network: Computation in Neural Systems*, 18(4):375–407, 2007.

Lasko, Thomas A. Efficient inference of Gaussian process modulated renewal processes with application to medical event data. In *Uncertainty in Artificial Intelligence*, 2014.

Lian, Wenzhao, Rao, Vinayak, Eriksson, Brian, and Carin, Lawrence. Modeling correlated arrival events with latent semi-markov processes. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 396–404, 2014.

Lloyd, Chris, Gunter, Tom, Osborne, Michael A., and Roberts, Stephen J. Variational inference for Gaussian process modulated Poisson processes. *arXiv:1411.0254*, 2014.

Pillow, Jonathan W, Shlens, Jonathon, Paninski, Liam, Sher, Alexander, Litke, Alan M, Chichilnisky, EJ, and Simoncelli, Eero P. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454(7207):995–999, 2008.

Rad, Kamiar R. and Paninski, Liam. Information rates and optimal decoding in large neural populations. In *Advances in Neural Information Processing Systems*, pp. 846–854, 2011.

Rajaram, Shyamsundar, Graepel, Thore, and Herbrich, Ralf. Poisson-networks: A model for structured point processes. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pp. 277–284, 2005.

Rao, Vinayak and Teh, Yee Whye. Gaussian process modulated renewal processes. In *Advances in Neural Information Processing Systems*, pp. 2474–2482, 2011.

Snelson, Ed and Ghahramani, Zoubin. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pp. 1257–1264, 2006.

Titsias, Michalis. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009.

Weiss, Jeremy C. and Page, David. Forest-based point process for event prediction from electronic health records. In *Machine Learning and Knowledge Discovery in Databases*, pp. 547–562. 2013.

World Health Organization, Geneva. International classification of diseases, 10th revision (ICD-10). 2010.

Xu, Lizhen, Duan, Jason A, and Whinston, Andrew. Path to purchase: A mutually exciting point process model for online advertising and conversion. *Management Science*, 2014.

Yu, Kai, Tresp, Volker, and Schwaighofer, Anton. Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd International Conference on Machine Learning*, pp. 1012–1019, 2005.