Technometrics
A Journal of Statistics for the Physical, Chemical, and Engineering Sciences

Technometrics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/utch20

Joint Models for Event Prediction From Time Series and Survival Data

Xubo Yue & Raed Al Kontar

To cite this article: Xubo Yue & Raed Al Kontar (2020): Joint Models for Event Prediction From Time Series and Survival Data, Technometrics, DOI: 10.1080/00401706.2020.1832582

To link to this article: https://doi.org/10.1080/00401706.2020.1832582

TECHNOMETRICS
2020, VOL. 00, NO. 0, 1–10
https://doi.org/10.1080/00401706.2020.1832582

Taylor & Francis
Taylor & Francis Group

Joint Models for Event Prediction From Time Series and Survival Data

Xubo Yue and Raed Al Kontar

Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI

ABSTRACT
We present a nonparametric prognostic framework for individualized event prediction based on joint modeling of both time series and time-to-event data. Our approach exploits a multivariate Gaussian convolution process (MGCP) to model the evolution of time series signals and a Cox model to map time-to-event data with time series data modeled through the MGCP. Taking advantage of the unique structure imposed by convolved processes, we provide a variational inference framework to simultaneously estimate parameters in the joint MGCP-Cox model. This significantly reduces computational complexity and safeguards against model overfitting. Experiments on synthetic and real world data show that the proposed framework outperforms state-of-the art approaches built on two-stage inference and strong parametric assumptions. Technical details are available in the supplementary materials.

ARTICLE HISTORY
Received January 2020
Accepted September 2020

KEYWORDS
Condition monitoring signals; Gaussian convolution process; Remaining useful life; Survival model

1. Introduction

1.1. Background and Motivation

In recent years, the multivariate Gaussian process (MGP) has drawn significant attention as an efficient nonparametric approach to predict time series signal trajectories (Dürichen et al. 2015; Kontar et al. 2018b; Moreno-Muñoz, Artés-Rodríguez, and Álvarez 2018; Yue and Kontar 2020). The MGP draws its roots from multitask learning where transfer of knowledge is achieved through a shared representation between training and testing signals. One neat approach that achieves this knowledge transfer, employs convolution processes to construct the MGP. Specifically, each signal is expressed as a convolution of latent functions drawn from a Gaussian process (GP). Commonalities among training and testing signals are then captured by sharing these latent functions across the outputs (Álvarez et al. 2010; Titsias and Lawrence 2010; Álvarez and Lawrence 2011). Consequently, the multiple signals can be expressed as a single output from a common multivariate Gaussian convolution process (MGCP). Indeed, many recent studies have demonstrated the MGCP ability to account for nontrivial commonalities in the data and provide accurate predictive results (Zhao and Sun 2016; Cheng 2018; Guarnizo and Álvarez 2018; Yue and Kontar 2019a).

In this article, we explore the following question: can we use both survival data along with time series signals to obtain reliable event prediction? This is illustrated in Figure 1. As shown in the figure, our goal is to use both survival data and time series signals from training units to predict the survival probability and survival time of a partially observed testing unit. Naturally, the aforementioned question is often encountered in a wide range of applications, including: disease prognosis in clinical trials, event prediction using vital health signals from monitored patients at risk, remaining useful life estimation of operational units/machines and failure prognosis in connected manufacturing systems (e.g., nuclear power plants) (Tsiatis, Degruttola, and Wulfsohn 1995; Gasmi, Love, and Kahle 2003; Pham, Yang, and Nguyen 2012; Gao et al. 2015; Soleimani, Hensman, and Saria 2018; Yue and Kontar 2019b).

To link survival and time series data, state-of-the-art methods have focused on joint models. The seminal work of Rizopoulos (2011, 2012) laid a foundation for joint models where a linear mixed effects model is used to model time series signals. The coefficients of the mixed model are then used in a Cox model to compute the probability of event occurrence conditioned on the observed time series signals. This idea provided the bases for many extensions and applications in the literature (Crowther, Abrams, and Lambert 2012; Zhu et al. 2012; Crowther, Abrams, and Lambert 2013; Proust-Lima et al. 2014; He et al. 2015; Rizopoulos, Molenberghs, and Lesaffre 2017; Mauff et al. 2018). It is important to note here that joint methods are in general built using a two-stage inference procedure due to the joint-likelihood intractability and huge computational complexity. In two-stage inference, features from the time series data are first learned, these estimated features are then inserted into a survival model to predict event probabilities. Such an approach induces bias and fails to handle missing data and noisy observations. Despite that, some articles have shown that the two-stage procedure can produce competitive predictive results (Wulfsohn and Tsiatis 1997; Yu et al. 2004; Zhou et al. 2014; Mauff et al. 2018). Nevertheless, the foregoing works are based on strong parametric assumptions where signals are assumed to follow a specific parametric form and all the signals (training and testing) exhibit that same functional form. In other words, signals behave according to a similar trend but at different rates (i.e., different parameter values). This focus on parametricity is mainly driven by the same reasons to that

CONTACT Raed Al Kontar alkontar@umich.edu Industrial & Operations Engineering, University of Michigan, Ann Arbor, MI 48109.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/TECH.

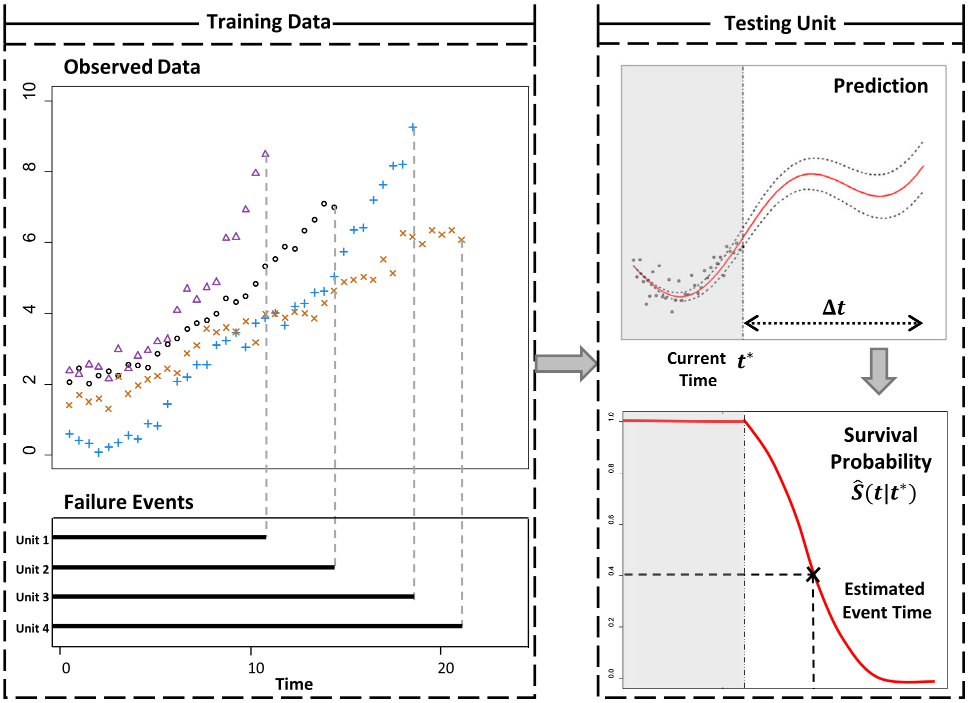

Figure 1. Joint modeling of longitudinal and time-to-event data.

of two-stage inference: joint-likelihood intractability and huge computational complexity.

Unfortunately, parametric methods are restrictive in many applications and if the specified form is far from the truth, predictive results will be misleading. Furthermore, the assumption that all signals possess the same functional form may not hold in real-life applications. For instance, units operated under different environmental conditions may exhibit different signal evolution rates and trends (Yan et al. 2016; Kontar et al. 2018a). Some recent efforts aimed to relax strong parametric assumptions using splines and continuous time Markov chains. Yet, these methods still assume homogeneity across the population and focus on merely imputing the time series data rather than predicting signal evolution within a time interval of interest (Dempsey et al. 2017; Soleimani, Hensman, and Saria 2018).

Inspired by recent advances in multi-output/MGPs, we propose a joint model that can overcome the aforementioned challenges. Indeed, MGPs have recently seen many success stories in the machine learning (ML) community. This success can be largely attributed to two advances: (i) the convolution construction (van der Wilk, Rasmussen, and Hensman 2017) of GPs which has enabled accounting for heterogeneity and nontrivial commonalities in outputs (here outputs refer to the time series signals); (ii) the variational inference framework (Snelson and Ghahramani 2006; Damianou and Lawrence 2013) which has enabled GPs to scale efficiently while regularizing inference to avoid overfitting.

Exploiting these advances, we propose a joint modeling approach, denoted as MGCP-Cox, which exploits the unique structure imposed by convolved processes to seamlessly integrate the Cox and MGP model into a unified framework for predicting survival times and conditional survival probabilities. A key interesting finding is that the marriage of MGCP and Cox results in a tractable variational likelihood which in turn allows simultaneous estimation of the joint model parameters. Based on this, we then derive event occurrence probabilities within any future interval Δt (as shown in Figure 1) and survival times. Using synthetic and real-world data we show the advantages properties of MGCP-Cox over several state-of-the-art techniques.

1.2. Related Work

An alternative method for joint analysis of time series data and survival data are treating it as a functional classification problem. In such a setting, the time series data are the functional input and the survival output within the interval time Δt is the response. For example, Alaa, Hu, and van der Schaar (2017) proposed a semi-Markov-modulated process to compute risk score of patient based on the absorbing probabilities. This score can assist doctors in building treatment plans. Futoma, Hariharan, and Heller (2017) modeled time series data using multitask GP and then used the results in a recurrent neural network classifier to detect Sepsis. The covariance structure used in this model is separable which, unlike convolved processes, fails to account for data heterogeneity (van der Wilk, Rasmussen, and Hensman 2017). Besides, this classification approach is a black box method that poses drawbacks similar to that of two-stage inference where they fail to account for missingess, uncertainty and noisy data. Further, such methods only provide survival probabilities within Δt and fall short of estimating survival times which are crucial in many applications. As we will show in the experiment part, the joint model is advantageous to classification approaches.

Here, we also distinguish joint models from the many works that model the intensity function of a point process (e.g., Cox, Poisson) using a GP or neural networks (Cunningham, Shenoy, and Sahani 2008; Lloyd et al. 2015; Alaa and van der Schaar 2017; Mei and Eisner 2017; Xiao et al. 2017). In the Cox process, for example, people are interested in modeling the arrival rate (e.g., customer arrival rate, disease recurrence rate) of a process. While in the joint model, we are estimating the survival probability/time using time series and survival data. We also note that there has been some recent attempts at rebuilding the Cox model using a GP (Fernández, Rivera, and Teh 2016; Kim and Pavlovic 2018). Kim and Pavlovic (2018) focused on the survival data and use GP to model the hazard function of Cox model, while Fernández, Rivera, and Teh (2016) used GP to model variations in the baseline hazard function. However, these approaches are only based on survival data and do not handle joint modeling, which is the focus of this article.

The rest of the article is organized as follows. In Section 2, we review survival analysis. In Section 3, we present our joint modeling framework and inference algorithm. Numerical experiments are provided in Section 4. Finally, Section 5 concludes the article with a brief discussion. A detailed code is deferred to the supplementary materials.

2. Review on Survival Analysis

In this section, we will briefly review survival analysis which will be used for event prediction in the joint model. Survival analysis is a branch of statistics for analyzing survival data and predicting the probability of occurrence of an event. For each individual unit i, the associated data are $\mathcal{D}_i = (V_i, \delta_i, \boldsymbol{Y}_i, \boldsymbol{w}_i)$, where $V_i = \min\{T_i, C_i\}$ is the event time (the unit failed at time T_i or was censored at time C_i), $\delta_i \in \{0, 1\}$ is an event indicator ($\delta_i = 1/0$ indicates the unit has failed/censored), $\boldsymbol{Y}_i$ are the noisy observed time series data (e.g., vital signals collected from patients) corresponding to the underlying latent values $\boldsymbol{f}_i$, and $\boldsymbol{w}_i$ is a set of time-invariant features (e.g., patient's gender). Typically, the continuous random variable T_i is characterized by a survival function $S(t) = P(T \geq t)$ which represents the probability of survival up to time t. Another important term is the hazard function $h(t) = \lim_{\Delta\to 0} \frac{1}{\Delta} P(t < T \leq t + \Delta | T \geq t) = -\frac{d}{dt} \log S(t)$ and can be thought of as the instantaneous rate of occurrence of an event at time t. It is easy to show that $S(t) = \exp\{-\int_0^t h(u)du\}$. The term $\int_0^t h(u)du$ is called cumulative hazard function and is denoted by $H(t)$. The basic scheme of survival analysis is to find suitable models to explain relationships between the hazard function $h_i(t)$ and collected data $\mathcal{D}_i$. These models are defined as survival models. Numerous survival models have been developed to analyze survival data. They typically model the hazard function as a function of some time-varying and fixed features. One class of prevailed survival models is called the Cox model (Cox 1972), which has the form $h_i(t) = h_0(t) \exp[\boldsymbol{\gamma}^T \boldsymbol{w}_i + \beta f_i(t)]$, where $h_0(t)$ is a baseline hazard function shared by all individuals, and is typically modeled by the Weibull or a piecewise constant function, $\boldsymbol{\gamma}$ is a vector of coefficients for the fixed covariates (features), $f_i(t)$ is the feature estimated by a time series model (e.g., linear mixed model, GP), and β is a scaling parameter for the time-varying covariates. Parameters in the Cox model are typically estimated by maximizing the full log-likelihood function $\sum_{i=1}^N \log p(V_i, \delta_i | \boldsymbol{w}_i, \boldsymbol{f}_i)$ defined as

$$\sum_{i=1}^{N} \{\delta_i \log \left[h_0(V_i) \exp[\boldsymbol{\gamma}^T \boldsymbol{w}_i + \beta f_i(V_i)]\right] - \int_0^{V_i} h_0(u) \exp[\boldsymbol{\gamma}^T \boldsymbol{w}_i + \beta f_i(u)] du\}. \quad (1)$$

For a comprehensive review of survival models, see Kalbfleisch and Prentice (2011).

Given an estimate of parameters from the Cox model, we can then obtain the event (failure) probability within a future time interval Δt given the fact that the testing unit i survives non-shorter than the current time instance t^*. This probability, denoted $\hat{P}_{\Delta t}$, is estimated as follows:

$$\begin{aligned} \hat{P}_{\Delta t} &= 1 - \hat{S}(t^* + \Delta t | t^*, \boldsymbol{w}_i, \boldsymbol{f}_i) = 1 - \frac{\hat{S}(t^* + \Delta t | \boldsymbol{w}_i, \boldsymbol{f}_i)}{\hat{S}(t^* | \boldsymbol{w}_i, \boldsymbol{f}_i)} \\ &= 1 - \exp\Big\{ - \int_{t^*}^{t^* + \Delta t} \hat{h}_0(u) \exp\big[\hat{\boldsymbol{\gamma}}^T \boldsymbol{w}_i + \hat{\beta} f_i(u)\big] du\Big\}, \end{aligned} \quad (2)$$

where $\boldsymbol{w}_i$ and $\boldsymbol{f}_i$ are features for a testing unit i. Note that in Figure 1 we show the survival curve which is defined as $\hat{S}(t|t^*) = 1 - \hat{P}_{\Delta t}$, where $t = t^* + \Delta t$. As shown in (2), event prediction requires predicting $f(u)$ within interval Δt. Indeed for the testing unit this requires extrapolation which is typically hard for nonparametric methods. However, a key feature of multitask approaches like the MGCP is that extrapolation implies interpolation across the testing and training signals while weighting the effect of different training signals on the test signal. Naturally then for nonparametric approaches, we require a training dataset that can capture different underlying behaviors and provide coverage throughout the sampling domain in which the experiment is performed.

3. Joint Modeling and Variational Inference

3.1. Setting

Assume data have been collected from N units and let $\mathcal{I} = \{1, 2, \ldots, N\}$ denote the set of all units. For unit i, its associated data are $\mathcal{D}_i = \{V_i, \delta_i, \boldsymbol{Y}_i, \boldsymbol{w}_i\}$. The observed time series signal is denoted by $\boldsymbol{Y}_i = (y_i(t_{i1}), \ldots, y_i(t_{il_i}))^T$, where l_i represents the number of observations and $\{t_{ir} : r = 1, \ldots, l_i\}$ denotes the inputs. We decompose the time series signal as $y_i(t) = f_i(t) + \epsilon_i(t)$, where $f_i(\cdot)$ is a mean zero GP and $\epsilon_i(t)$ denotes additive noise with zero mean and σ_ϵ^2 variance. Without loss of generality, assume unit $1, \ldots, N-1$ are training units and unit N is the testing unit. Our goal is to predict the survival probability of testing unit N. Note that our joint model is also capable of handling multiple testing units. However, for simplicity, we only focus on a single testing unit in the remaining of this article. Throughout this article, we use $p(\cdot)$ to represent the probability density function of a random variable and use $\phi(\cdot|\boldsymbol{a}, \boldsymbol{A})$ to denote a normal density function of a random variable with mean vector $\boldsymbol{a}$ and covariance matrix $\boldsymbol{A}$.

3.2. The Multivariate Gaussian Convolution Process (MGCP)

To obtain an accurate predictive result, we need to capture the intrinsic relatedness among N signals. Particularly, we resort to the convolution process to model the latent function $f_i(t)$. We consider K independent latent functions $\{X_k(t)\}_{k=1}^K$ and NK different smoothing kernels $\{G_{ik}(t) : i \in \mathcal{I}\}_{k=1}^K$. The latent functions are assumed independent GPs with covariance $\text{cov}[X_k(t), X_k(t')] = \tau_k(t,t')$. We set $G_{ik}(t) = \alpha_{ik}\phi(t|0,\xi_{ik}^2) := \alpha_{ik}\frac{1}{\sqrt{2\pi\xi_{ik}^2}}\exp(-\frac{t^2}{2\xi_{ik}^2})$ to be scaled Gaussian kernels and $\tau_k(t,t')$ to be squared exponential covariance functions (Álvarez and Lawrence 2009).

$$\tau_k(t,t') = \exp\left[-\frac{1}{2}\frac{(t-t')^2}{\lambda_k^2}\right]. \tag{3}$$

Note that the choice of latent functions and smoothing kernels will not affect the inference procedure. Practitioners can replace them with any popular choices based on the domain knowledge.

The GP $f_i(t)$ is then constructed by convolving the shared latent functions with the smoothing kernel as shown in (4). This is the underlying principle of the MGCP, where the latent functions $\{X_k(t)\}_{k=1}^K$ are shared across different outputs through the corresponding kernels $G_{ik}(t)$. Since convolutions are linear operators on a function and since the latent function, a GP, is shared across multiple outputs then all outputs can be expressed as a jointly distributed GP, an MGCP. A key feature is that information is shared through different parameters encoded in the kernels $G_{ik}(t)$. Outputs then can possess both shared and unique features, accounting for heterogeneity in the time series data.

$$f_i(t) = \sum_{k=1}^K \int_{\mathbb{R}} G_{ik}(t-u)X_k(u)du. \tag{4}$$

Based on Equation (4), the covariance function between f_i and f_j, and the covariance function between f_i and X_k, can be calculated in closed forms. Please refer to Appendix A in the supplementary materials for details. Now denote the underlying latent values as $\boldsymbol{f} = \{\boldsymbol{f}_1^T, \dots, \boldsymbol{f}_N^T\}^T$, where $\boldsymbol{f}_i = \{f_i(t_{i1}), \dots, f_i(t_{il_i})\}^T$. The density function of $\boldsymbol{f}$ can be obtained as $p(\boldsymbol{f}) = \phi(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K_{ff}})$, where $|\cdot|$ is the determinant and $\boldsymbol{K_{ff}}$ sized $(\sum_{i=1}^N l_i) \times (\sum_{i=1}^N l_i)$ is the covariance function. The likelihood of $\boldsymbol{f}$ involves inverting the large matrix $\boldsymbol{K_{ff}}$. This operation has computational complexity $\mathcal{O}((\sum_{i=1}^N l_i)^3)$ and storage requirement $\mathcal{O}((\sum_{i=1}^N l_i)^2)$. To alleviate computational burden, we use the inducing variable approximation (van der Wilk, Rasmussen, and Hensman 2017) which relies on M pseudo-inputs from the latent functions denoted as $\boldsymbol{X}_k(Z) = [X_k(z_1), \dots, X_k(z_M)]^T$ where $Z = \{z_i\}_{i=1}^M$. Since the latent functions are GPs, then any sample $\boldsymbol{X}_k(Z)$ follows a multivariate Gaussian distribution. Conditioned on $\boldsymbol{X}_k(Z)$, we next sample from the conditional prior $p(X_k(u)|\boldsymbol{X}_k(Z))$. In Equation (4), $X_k(u)$ can be approximated well by the expectation $\mathbb{E}(X_k(u)|\boldsymbol{X}_k(Z))$ as long as the latent functions are smooth (Álvarez and Lawrence 2011). Here, we note that in the context of GPs, this is known as the fully training independent conditional (FITC) approximation (Quiñonero-Candela and Rasmussen 2005). Now, denote by $\boldsymbol{X} = \{\boldsymbol{X}_1^T(Z), \dots, \boldsymbol{X}_K^T(Z)\}^T$. The probability distribution of $\boldsymbol{X}$ can be expressed as $p(\boldsymbol{X}|Z) = \phi(\boldsymbol{X}|\boldsymbol{0}, \boldsymbol{K_{XX}})$, where $\boldsymbol{K_{XX}}$ is a block-diagonal matrix such that each block is associated with the covariance of X_k in (3). By multivariate Gaussian identities, the probability distribution of $\boldsymbol{f}$ conditional on $\boldsymbol{X}, Z$ is

$$p(\boldsymbol{f}|\boldsymbol{X}, Z) = \phi(\boldsymbol{f}|\boldsymbol{K_{fX}}\boldsymbol{K_{XX}^{-1}}\boldsymbol{X}, \boldsymbol{K_{ff}} - \boldsymbol{Q}), \tag{5}$$

where $\boldsymbol{Q} = \boldsymbol{K_{fX}}\boldsymbol{K_{XX}^{-1}}\boldsymbol{K_{Xf}}$. Therefore, $p(\boldsymbol{f})$ can be approximated by $p(\boldsymbol{f}|Z)$, which is given as

$$p(\boldsymbol{f}|Z) = \int p(\boldsymbol{f}|\boldsymbol{X}, Z)p(\boldsymbol{X}|Z)d\boldsymbol{X}. \tag{6}$$

From (6), $p(\boldsymbol{Y})$ can be obtained by $p(\boldsymbol{Y}|Z) = \int p(\boldsymbol{Y}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{X}, Z)p(\boldsymbol{X}|Z)d\boldsymbol{f}d\boldsymbol{X}$.

3.3. Joint Model and Variational Inference

Now following our convolution construction in (4), the hazard function at time t is given as

$$h_i(t) = h_0(t)\exp\left[\boldsymbol{\gamma}^T\boldsymbol{w}_i + \beta\sum_{k=1}^K \int_{\mathbb{R}} G_{ik}(t-u)X_k(u)du\right]. \tag{7}$$

This key equation links the MGCP to the Cox model. We begin with presenting the log-likelihood of the joint model given observed data $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\mathcal{D}}_i\}_{i=1}^N$. The marginal log-likelihood function is

$$\begin{aligned}\log p(\boldsymbol{\mathcal{D}}) &= \log\int p(\boldsymbol{\mathcal{D}}|\boldsymbol{f})p(\boldsymbol{f})d\boldsymbol{f} = \log\int p(\boldsymbol{\mathcal{D}}|\boldsymbol{f})p(\boldsymbol{f}|Z)d\boldsymbol{f}\\ &= \log\int p(\boldsymbol{\mathcal{D}}|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{X}, Z)p(\boldsymbol{X}|Z)d\boldsymbol{X}d\boldsymbol{f}.\end{aligned} \tag{8}$$

We would like to provide a good approximation of $\log p(\boldsymbol{\mathcal{D}})$ by introducing an evidence lower bound (ELBO) $\mathcal{L}$. This bound is calculated by finding the Kullback–Leibler (KL) divergence between the variational density $q(\boldsymbol{f}, \boldsymbol{X}|Z)$ and the true posterior density $p(\boldsymbol{f}, \boldsymbol{X}|\boldsymbol{\mathcal{D}}, Z)$. Specifically,

$$\begin{aligned}&\text{KL}(q(\boldsymbol{f}, \boldsymbol{X}|Z)||p(\boldsymbol{f}, \boldsymbol{X}|\boldsymbol{\mathcal{D}}, Z))\\ &= \int q(\boldsymbol{f}, \boldsymbol{X}|Z)\log\frac{q(\boldsymbol{f}, \boldsymbol{X}|Z)}{p(\boldsymbol{f}, \boldsymbol{X}|\boldsymbol{\mathcal{D}}, Z)}d\boldsymbol{X}d\boldsymbol{f}\\ &= \int q(\boldsymbol{f}, \boldsymbol{X}|Z)\log\frac{q(\boldsymbol{f}, \boldsymbol{X}|Z)p(\boldsymbol{\mathcal{D}})}{p(\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{\mathcal{D}}|Z)}d\boldsymbol{X}d\boldsymbol{f}\\ &= \log p(\boldsymbol{\mathcal{D}}) - \int q(\boldsymbol{f}, \boldsymbol{X}|Z)\log\frac{p(\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{\mathcal{D}}|Z)}{q(\boldsymbol{f}, \boldsymbol{X}|Z)}d\boldsymbol{X}d\boldsymbol{f}\\ &= \log p(\boldsymbol{\mathcal{D}}) - \mathcal{L} \geq 0.\end{aligned} \tag{9}$$

The variational density is assumed to be factorized as

$$q(\boldsymbol{f}, \boldsymbol{X}|Z) = p(\boldsymbol{f}|\boldsymbol{X}, Z)q(\boldsymbol{X}). \tag{10}$$

Maximizing the ELBO with respect to $q(\boldsymbol{X})$ and hyperparameters from the MGCP-Cox model can achieve purposes of variational inference and model selection simultaneously. By Equation (9),

$$\begin{aligned}\mathcal{L} &= \int q(\boldsymbol{f},\boldsymbol{X}|Z)\log\frac{p(\boldsymbol{f},\boldsymbol{X},\boldsymbol{\mathcal{D}}|Z)}{q(\boldsymbol{f},\boldsymbol{X}|Z)}d\boldsymbol{X}d\boldsymbol{f}\\ &= \int q(\boldsymbol{X})\int p(\boldsymbol{f}|\boldsymbol{X},Z)\log p(\boldsymbol{\mathcal{D}}|\boldsymbol{f})d\boldsymbol{f}d\boldsymbol{X} \\ &\quad+\int q(\boldsymbol{X})\log\frac{p(\boldsymbol{X}|Z)}{q(\boldsymbol{X})}d\boldsymbol{X}.\end{aligned} \tag{11}$$

Furthermore, we can decompose $\log p(\boldsymbol{\mathcal{D}}|\boldsymbol{f}) = \log p(\boldsymbol{Y}|\boldsymbol{f}) + \log p(\boldsymbol{V},\boldsymbol{\delta}|\boldsymbol{w},\boldsymbol{f})$, where $\boldsymbol{V} = \{V_i\}_{i=1}^N$, $\boldsymbol{\delta} = \{\delta_i\}_{i=1}^N$ and $\boldsymbol{w} = \{w_i\}_{i=1}^N$. Based on Equation (11), the MGCP propagates uncertainties through the latent processes to the Cox model.

It is desirable to find a closed form of the ELBO in Equation (11). Since $p(\boldsymbol{Y}|\boldsymbol{f})$ and $p(\boldsymbol{f}|\boldsymbol{X},Z)$ are both Gaussian, we can obtain

$$\begin{aligned}\int p(\boldsymbol{f}|\boldsymbol{X},Z)\log p(\boldsymbol{Y}|\boldsymbol{f})d\boldsymbol{f} &= \log\phi(\boldsymbol{Y}|\boldsymbol{K}_{fX}\boldsymbol{K}_{XX}^{-1}\boldsymbol{X},\sigma_\epsilon^2\boldsymbol{I})\\ &\quad-\frac{1}{2\sigma_\epsilon^2}\mathrm{Tr}(\boldsymbol{K}_{ff}-\boldsymbol{Q}),\end{aligned} \tag{12}$$

where $\mathrm{Tr}(\cdot)$ is a trace operator. Therefore, the ELBO can be simplified as

$$\begin{aligned}\mathcal{L} &= -\frac{1}{2\sigma_\epsilon^2}\mathrm{Tr}(\boldsymbol{K}_{ff}-\boldsymbol{Q})\\ &\quad+\int q(\boldsymbol{X})\log\frac{\phi(\boldsymbol{Y}|\boldsymbol{K}_{fX}\boldsymbol{K}_{XX}^{-1}\boldsymbol{X},\sigma_\epsilon^2\boldsymbol{I})p(\boldsymbol{X}|Z)}{q(\boldsymbol{X})}d\boldsymbol{X}\\ &\quad+\int q(\boldsymbol{X})p(\boldsymbol{f}|\boldsymbol{X},Z)\log p(\boldsymbol{V},\boldsymbol{\delta}|\boldsymbol{w},\boldsymbol{f})d\boldsymbol{f}d\boldsymbol{X}.\end{aligned} \tag{13}$$

We compute the optimal upper bound of $\mathcal{L}$ by reversing Jensen's inequality. This gives an optimal distribution $q^*(\boldsymbol{X})$ and

$$\begin{aligned}\mathcal{L}^* &= \log\int\phi(\boldsymbol{Y}|\boldsymbol{K}_{fX}\boldsymbol{K}_{XX}^{-1}\boldsymbol{X},\sigma_\epsilon^2\boldsymbol{I})p(\boldsymbol{X}|Z)d\boldsymbol{X}+\boldsymbol{PE}\\ &\quad+\int q(\boldsymbol{X})p(\boldsymbol{f}|\boldsymbol{X},Z)\log p(\boldsymbol{V},\boldsymbol{\delta}|\boldsymbol{w},\boldsymbol{f})d\boldsymbol{f}d\boldsymbol{X}\\ &= \log[\phi(\boldsymbol{Y}|\boldsymbol{0},\sigma_\epsilon^2\boldsymbol{I}+\boldsymbol{Q})]+\boldsymbol{PE}\\ &\quad+\int q(\boldsymbol{X})p(\boldsymbol{f}|\boldsymbol{X},Z)\log p(\boldsymbol{V},\boldsymbol{\delta}|\boldsymbol{w},\boldsymbol{f})d\boldsymbol{f}d\boldsymbol{X},\end{aligned} \tag{14}$$

where $\boldsymbol{PE} = -\frac{1}{2\sigma_\epsilon^2}\mathrm{Tr}(\boldsymbol{K}_{ff}-\boldsymbol{Q})$. $\boldsymbol{PE}$ can be thought of as a penalization term that regularizes the estimation of the parameters. Note that the first two terms in Equation (14) can be computed in $\mathcal{O}((\sum_{i=1}^N l_i)M^2)$ (Snelson and Ghahramani 2006).

3.4. Variational Inference on Cox Model

Parameters in the Cox model can be attained by maximizing the following log-likelihood function:

$$\begin{aligned}\log p(\boldsymbol{V},\boldsymbol{\delta}|\boldsymbol{w},\boldsymbol{f}) &= \sum_{i=1}^N\log p(V_i,\delta_i|\boldsymbol{w}_i,\boldsymbol{f}_i)\\ &= \sum_{i=1}^N\Big\{\delta_i\log\big[h_0(V_i)\exp[\boldsymbol{\gamma}^T\boldsymbol{w}_i\\ &\quad+\beta\sum_{k=1}^K\int_{\mathbb{R}}G_{ik}(V_i-u)X_k(u)du]\big]\\ &\quad-\int_0^{V_i}h_0(u)\exp[\boldsymbol{\gamma}^T\boldsymbol{w}_i\\ &\quad+\beta\sum_{k=1}^K\int_{\mathbb{R}}G_{ik}(u-v)X_k(v)dv]du\Big\}.\end{aligned} \tag{15}$$

In Equation (14), we obtain the optimal $q^*(\boldsymbol{X})$ to maximize the ELBO. In this section, we will use it. Specifically, the optimal $q^*(\boldsymbol{X})$ has the form

$$\begin{aligned}&q^*(\boldsymbol{X}) = \phi(\boldsymbol{X}|\sigma_\epsilon^{-2}\boldsymbol{K}_{XX}(\boldsymbol{K}_{XX}+\sigma_\epsilon^{-2}\boldsymbol{K}_{Xf}\boldsymbol{K}_{fX})^{-1}\boldsymbol{K}_{Xf}\boldsymbol{Y},\\ &\boldsymbol{K}_{XX}(\boldsymbol{K}_{XX}+\sigma_\epsilon^{-2}\boldsymbol{K}_{Xf}\boldsymbol{K}_{fX})^{-1}\boldsymbol{K}_{XX}) \coloneqq \phi(\boldsymbol{X}|\boldsymbol{m},\boldsymbol{s}).\end{aligned} \tag{16}$$

It is easy to show that $q(\boldsymbol{f}|Z)$ has the normal distribution with parameter $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$. Specifically,

$$\int q^*(\boldsymbol{X})p(\boldsymbol{f}|\boldsymbol{X},Z)d\boldsymbol{X} = q(\boldsymbol{f}|Z)\coloneqq q(\boldsymbol{f}),\ \boldsymbol{f}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma}), \tag{17}$$

where

$$\boldsymbol{\mu} = \boldsymbol{K}_{fX}\boldsymbol{K}_{XX}^{-1}\boldsymbol{m},\quad \boldsymbol{\Sigma} = \boldsymbol{K}_{ff}-\boldsymbol{K}_{fX}\boldsymbol{K}_{XX}^{-1}(\boldsymbol{I}-\boldsymbol{s}\boldsymbol{K}_{XX}^{-1})\boldsymbol{K}_{Xf}.$$

The last integration in Equation (14) can be simplified to

$$\begin{aligned}&\int q(\boldsymbol{f})\log p(\boldsymbol{V},\boldsymbol{\delta}|\boldsymbol{w},\boldsymbol{f})d\boldsymbol{f}\\ &= \int q(\boldsymbol{f})\sum_{i=1}^N\big\{\delta_i\log\big[h_0(V_i)\exp[\boldsymbol{\gamma}^T\boldsymbol{w}_i+\beta f_i(V_i)]\big]\\ &\quad-\int_0^{V_i}h_0(u)\exp[\boldsymbol{\gamma}^T\boldsymbol{w}_i+\beta f_i(u)]du\big\}d\boldsymbol{f}.\end{aligned} \tag{18}$$

The first term in Equation (18) can be calculated analytically. For each unit i,

$$\begin{aligned}&\int q(\boldsymbol{f})\delta_i\log[h_0(V_i)\exp[\boldsymbol{\gamma}^T\boldsymbol{w}_i+\beta f_i(V_i)]]d\boldsymbol{f}\\ &= \delta_i\big\{\log h_0(V_i)+\boldsymbol{\gamma}^T\boldsymbol{w}_i+\beta\mathbb{E}_{q(\boldsymbol{f})}[f_i(V_i)]\big\}\\ &= \delta_i\big\{\log h_0(V_i)+\boldsymbol{\gamma}^T\boldsymbol{w}_i+\beta\mu_i(V_i)\big\},\end{aligned} \tag{19}$$

where $\mathbb{E}_{q(\boldsymbol{f})}[f_i(V_i)] = \boldsymbol{K}_{f_i(V_i)X}\boldsymbol{K}_{XX}{}^{-1}\boldsymbol{m} \coloneqq \mu_i(V_i)$. The second term in Equation (18) can also be further simplified. For each unit i,

$$\begin{aligned}&\int q(\boldsymbol{f})(-\int_0^{V_i}h_0(u)\exp[\boldsymbol{\gamma}^T\boldsymbol{w}_i+\beta f_i(u)]du)d\boldsymbol{f}\\ &= -\int_0^{V_i}h_0(u)\exp[\boldsymbol{\gamma}^T\boldsymbol{w}_i]\exp\big[\beta[\mu_i(u)+\frac{1}{2}\sigma_i^2(u)]\big]du,\end{aligned} \tag{20}$$

where $\mu_i(u) = \boldsymbol{K}_{f_i(u)\boldsymbol{X}}\boldsymbol{K}_{\boldsymbol{XX}}{}^{-1}\boldsymbol{m}$ and $\sigma_i^2(u) = \boldsymbol{K}_{f_i(u)f_i(u)} - \boldsymbol{K}_{f_i(u)\boldsymbol{X}}\boldsymbol{K}_{\boldsymbol{XX}}^{-1}(\boldsymbol{I} - \boldsymbol{s}\boldsymbol{K}_{\boldsymbol{XX}}^{-1})\boldsymbol{K}_{\boldsymbol{X}f_i(u)}$. Please refer to Appendix B in the supplementary materials for details.

We can assume $h_0(t)$ to be an exponential function $\exp(b + \psi(t - \min\{V_i\}_{i=1}^N))$, where b,ψ are parameters to be learned and $h_0(t) = 0$ when $t < \min\{V_i\}_{i=1}^N$ because units are not subject to risk before the first failure event. Note that if we assume the baseline hazard is nondecreasing with time, we can add one constraint $\psi \in \mathbb{R}_+$. Otherwise, we can use $\psi \in \mathbb{R}$. We can also use smoothing spline to get a robust baseline hazard estimation. Please refer to Section 3.6 for details.

The $\mathcal{L}^*$ is maximized with respect to the parameters $\Theta = (\boldsymbol{\theta}, \sigma_\epsilon, \boldsymbol{\gamma}, \beta, b, \psi)$, where $\boldsymbol{\theta} = (\{\lambda_k, \xi_{ik}, \alpha_{ik}\}_{i=1,k=1}^{N,K})$, by the gradient-based method. Specifically, We can obtain the optimal parameters $\hat{\Theta}$ by maximizing $\mathcal{L}^*$.

3.5. *Event Prediction*

Without loss of generality, we focus on predicting the event occurrence probability for unit N. Suppose observations from the testing unit N have been collected up to time t^*. The survival model computes the event probabilities conditioned on the predicted time series features $\boldsymbol{f}_N(u), u \in [t^*, t^* + \Delta t]$. Given estimated parameters, and following (2), we are interested in calculating

$$\begin{aligned} 1 - \hat{S}(t^* + \Delta t | t^*, \boldsymbol{w}_N, \boldsymbol{f}_N) &= 1 - \frac{\hat{S}(t^* + \Delta t | \boldsymbol{w}_N, \boldsymbol{f}_N)}{\hat{S}(t^* | \boldsymbol{w}_N, \boldsymbol{f}_N)} \\ &= 1 - \exp\{-\int_{t^*}^{t^*+\Delta t} \hat{h}_0(u) \exp\left[\hat{\boldsymbol{\gamma}}^T \boldsymbol{w}_N + \hat{\beta} f_N(u)\right] du\}. \end{aligned} \tag{21}$$

Based on Equation (21), the accurate extrapolation within Δt is essential. In the MGCP, the predictive distribution for any new input point T is given by

$$\begin{aligned} p(f_N(T^*)|\boldsymbol{Y}) &= \int p(f_N(T^*)|\boldsymbol{X}) p(\boldsymbol{X}|\boldsymbol{Y}) d\boldsymbol{X} \\ &= \int \phi(f_N | \boldsymbol{K}_{f_N(T^*)\boldsymbol{X}} \boldsymbol{K}_{\boldsymbol{XX}}^{-1} \boldsymbol{X}, \boldsymbol{W}) p(\boldsymbol{X}|\boldsymbol{Y}) d\boldsymbol{X} \\ &= \int \phi(f_N | \boldsymbol{K}_{f_N(T^*)\boldsymbol{X}} \boldsymbol{K}_{\boldsymbol{XX}}^{-1} \boldsymbol{X}, \boldsymbol{W}) \frac{p(\boldsymbol{Y}|\boldsymbol{X}) p(\boldsymbol{X})}{p(\boldsymbol{Y})} d\boldsymbol{X} \\ &= \phi(f_N | \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{Y}, \boldsymbol{K}_{f_N(T^*)f_N(T^*)} - \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{A}^T), \end{aligned} \tag{22}$$

where

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{K}_{f_N(T^*)\boldsymbol{X}} \boldsymbol{K}_{\boldsymbol{XX}}^{-1} \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{f}}; \\ \boldsymbol{W} &= K_{f_N(T^*)f_N(T^*)} - \boldsymbol{K}_{f_N(T^*)\boldsymbol{X}} \boldsymbol{K}_{\boldsymbol{XX}}^{-1} \boldsymbol{K}_{\boldsymbol{X}f_N(T^*)}. \end{aligned}$$

We have used $K_{f_N(T^*)f_N(T^*)}$ as a notation to indicate when the covariance matrix is evaluated at the T^*. Consequently, the predicted signal at the time point T^* for unit N is $\hat{f}_N(T^*) = \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{Y}$. Besides survival probability, the survival time can also be estimated in closed form. We provide detailed information in Section 4.2.

3.6. *Practical Issues*

In this section, we will discuss some practical issues about the algorithm implementation.

- Smoothing spline: To obtain a good baseline hazard prediction given the estimated $\hat{b}, \hat{\psi}$, we can calculate the cumulative hazard at time point t as $H(t) = \sum_{u \in \mathcal{F}(t)} \hat{h}_0(u)$, for all t, where $\mathcal{F}(t) := \left\{\{V_{(1)}, V_{(2)}, \ldots, V_{(N-1)}\} \cup \{0, 1, 2, \ldots, V_{(N)}\}\right\} \cap [0, t]$, and $V_{(i)}$ is the ith smallest element in $\{V_i\}_{i=1}^N$. Then we fit a regularized smooth spline to $H(t)$ (Ruppert 2002). The predicted baseline hazard at $u \in [t^*, t^* + \Delta t^*]$ can be estimated by $\left.\frac{d\hat{H}(t)}{dt}\right|_{t=u}$ (Rosenberg 1995).

  Smoothing of the baseline hazard has been a common practice that was observed to enhance prediction accuracy (Rosenberg 1995). This practice has also been adopted in most of the well-known survival analysis code libraries. For example, if we assume the baseline hazard to be piecewise constant or some piecewise function, then the cumulative hazard is not smooth and we cannot get a continuous estimation of $h_0(t)$. However, if we assume a smooth parametric form of $h_0(t)$, then the spline is not necessary. Overall, the smoothing spline is an add-on approach that is commonly used to ensure the robustness of the estimation method (mainly smoothness of the estimated baseline hazard).
- Number of pseudo-inputs: It has been widely investigated in MGCPs and the recent work in Burt, Rasmussen, and Wilk (2019). For smooth kernel, it is in the form of $M = \mathcal{O}(\log n)$ where n is number of data points. The key advantage is that M is much smaller than the number of data points. This achieves the so-called "sparse approximation."
- Number of latent variables: This is still an open question. Intuitively, when training units display a strong heterogeneity, we require more latent variables (and thus heavier computational burden) (Zhao and Sun 2016). Practically, as we will demonstrate in the experiment section, only one latent variable is flexible enough to provide an accurate prediction.

4. Experiments

We conduct case studies to demonstrate the performance of our proposed methodology. Both synthetic and real-world data are used.

4.1. *Data Setting*

For the synthetic data, we assume that the underlying true path for unit i has the polynomial trajectory with individualized random effects. Specifically, $y_i(t) = \boldsymbol{z}^T(t)\boldsymbol{b}_i + \epsilon_i(t) = b_{i0} + b_{i1}t + b_{i2}t^2 + \epsilon_i(t)$, where $\epsilon_i(t) \sim \mathcal{N}(0, 0.1)$, $\boldsymbol{z}^T(t) = [1, t, t^2]$ and $\boldsymbol{b}_i = [b_{i0}, b_{i1}, b_{i2}]^T \sim \mathcal{N}(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b)$ with $\boldsymbol{\mu}_b = [2.5, 0.1, a]^T$ and $\boldsymbol{\Sigma}_b = \begin{bmatrix} 0.2 & -4e-4 & -8e-5 \\ -4e-4 & 3e-6 & 3e-7 \\ -e-5 & 3e-7 & 1e-7 \end{bmatrix}$ where $a \sim$ uniform$(0.003, 0.03)$. Without loss of generality, we assume that the time unit is month and that signals were obtained regularly at each month up to their failure or censoring time. For each unit, we specify a time-invariant feature $w_i \in \{0, 1\}$ generated by a Bernoulli distribution with $p = 0.5$. In the Cox model, we use the Weibull baseline hazard rate function $h_0(t) = \lambda\rho t^{\rho-1}$ with $\lambda = 0.001$ and $\rho = 1.05$. We generate the failure time T_i for each unit by rejection sampling using its probability density function $h_i(t)S_i(t)$. We set $\gamma = 0$ and $\beta = 0.5$. Also, we randomly select 5% of the units to be right censored. The number of units generated is $N = 20$ and the experiment is repeated for $Q = 100$ times.

For the real-world case study we use the C-MAPSS dataset provided by the National Aeronautics and Space Administration (NASA). The dataset contains failure time data of aircraft turbofan engines and degradation signals from informative sensors mounted on these engines. Note that in our analysis we standardize all sensor data. We refer readers to Saxena and Goebel (2008) for more details about the data. We also conduct another real-world case study using data from automotive lead-acid batteries. Similar to the NASA data, each battery has a degradation signal and its own failure time. The signal measurements are irregular with missing values. In either dataset, the training sample size is 100.

4.2. Baselines and Evaluations

We focus on predicting the event probability within a future time interval Δt. We consider $\Delta t = 12, 15, 20$ months in this simulation study. Prediction performance at varying time points t^* for the partially observed unit N is then reported. The time instant $t^* = \alpha T_N$ is defined as the α-observation percentile, where T_N is the failure time of unit N. The values of α are specified as 30%, 50%. Further, in our simulation studies, we benchmark our method with four other reference methods for comparison: (1) support vector machine (SVM) classifier: in the SVM, event data is transformed into binary labels $\delta_i = 1/0$ denoting whether units failed or not within the time interval $[t^*, \Delta t + t^*]$. The time-fixed covariate w_i and the last observed signal measurement at t^* are used as the model predictors. We use the radial basis kernel and determine parameters using 2-fold cross-validation on the training data. (2) The multi-task GP recurrent neural network (RNN) classifier (GP-RNN) (Futoma, Hariharan, and Heller 2017): this method exploits a RNN to provide a binary event outcome prediction. (3) The deep recurrent survival model (Deep-S) (Ren et al. 2019): this model also uses deep neural network to estimate survival probability. However, it does not use any information from time series data. (4) Parametric joint model (LMM-joint): we implement a state-of-the-art joint modeling algorithm using the linear mixed-effect model. The LMM-joint uses a general polynomial function whose corresponding degree is determined through an Akaike information criteria to model the signal path. Note that this framework estimates parameters from the mixed-effect model and the Cox model separately (Rizopoulos 2011; Zhou et al. 2014; Mauff et al. 2018). Regarding our MGCP-Cox model we set the number of pseudo-inputs to $M = 128$ (Burt, Rasmussen, and Wilk 2019) and the number of latent functions to $K = 1$. This setting is a commonly used setting for the MGCP (Álvarez and Lawrence 2011; Zhao and Sun 2016). The performance of each method is then assessed by the receiver operating characteristic (ROC) curve, which is a common diagnostic tool for binary classifier. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR). Predictive accuracy is then assessed through the area under the curve (AUC). The results from the synthetic data are shown in Figure 2. Due to poor performance of the SVM on $N = 20$, we also checked whether it can produce comparable results to the MGCP-Cox when $N = 200$. We denote this model as SVM-200.

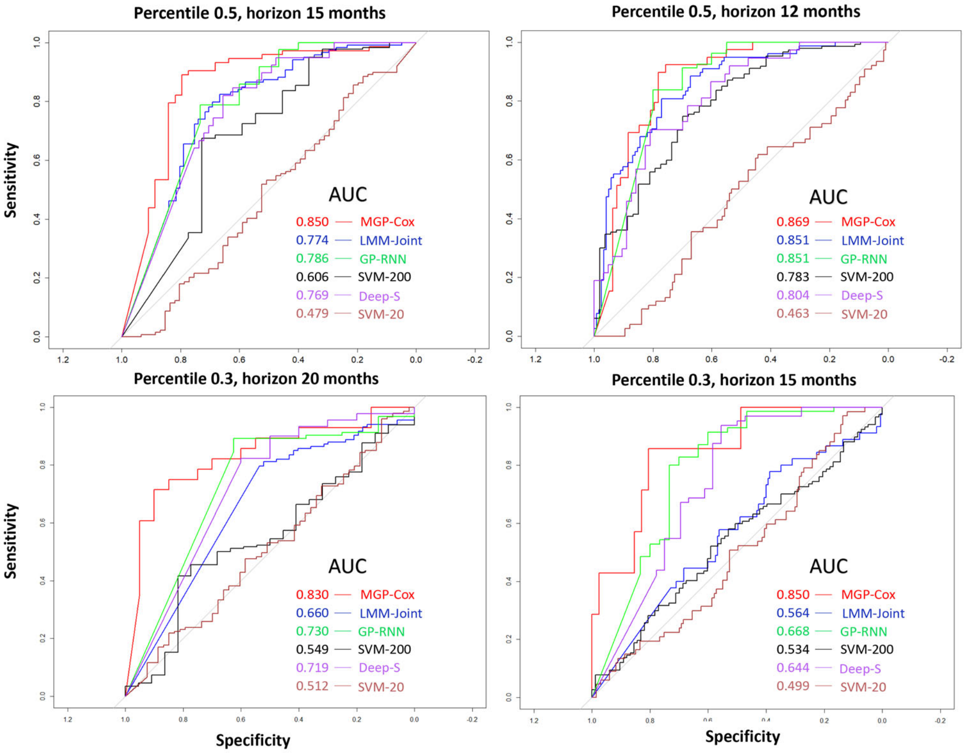

Figure 2. ROC curves from simulation studies under different percentile of observation α.

For the real data, the true survival probabilities are not available since we do not have information about the underlying parameters used to generate the data. Therefore, to evaluate model performance, we calculate the mean remaining lifetime (i.e., survival time) of the testing unit, which is defined as $\widehat{mrl}(t^*) = \int_{t^*}^{\infty} \hat{S}(u|t^*, \boldsymbol{w}_N, \boldsymbol{f}_N)du$. This integration can be obtained by the Gauss-Legendre quadrature. The performance is assessed by the absolute error $AE = |rl_j - \widehat{mrl}_j|$ where rl_j is the true remaining lifetime of the testing unit. We then report the distribution of the errors across all units using the boxplot in Figure 3. Similar to the synthetic data we use 30% and 50% percentiles to assess prediction accuracy. We also note that we cannot obtain $\widehat{mrl}$ estimates from all classification methods as they transform event prediction into a time series classification problem. Besides, there are very limited models which are capable of handling joint data and predicting survival time. Therefore, we only benchmark our model with (1) the state-of-the-art LMM-joint model. (2) The linear models of coregionalization (LMC) (Soleimani, Hensman, and Saria 2018). This method uses coregionalization rather than convolution to construct GP. Besides, this method ignores useful information from future evolution of time series signals. (3) the MGCP-Cox with two-stage inference (Joint-Two). In the Joint-Two, we apply the two-stage inference method. Therefore, the uncertainty in the time series data is not propagated via latent variables. All results are reported in Figures 3 and 4.

4.3. Results

The results are given in Figures 2–4. Based on the figures, we can obtain some important insights. First, the MGCP-Cox model clearly outperforms the benchmarked models and achieves high classification and prediction accuracy. This highlights the advantages of joint estimation compared to two-stage inference where time series uncertainty is not propagated to the survival model. Furthermore, the results highlight the advantages of the *joint model* compared to the *classification-based models* which are Deep-S, GP-RNN, and SVM. For SVM even with a much larger number of units, the MGCP-Cox was still superior.

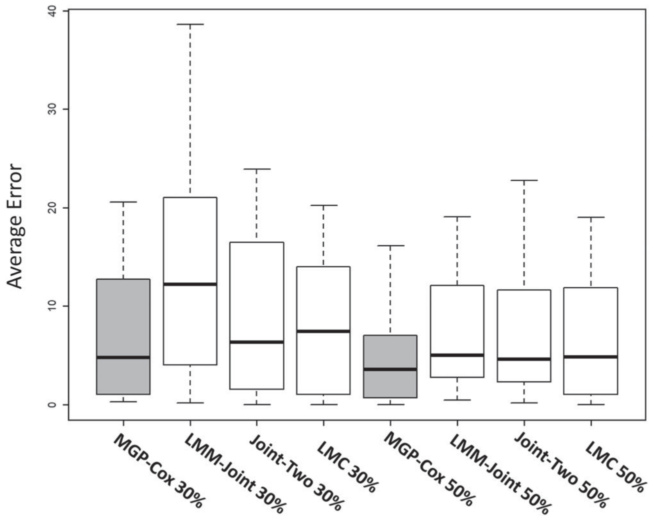

Figure 3. Remaining life prediction accuracy from NASA data.

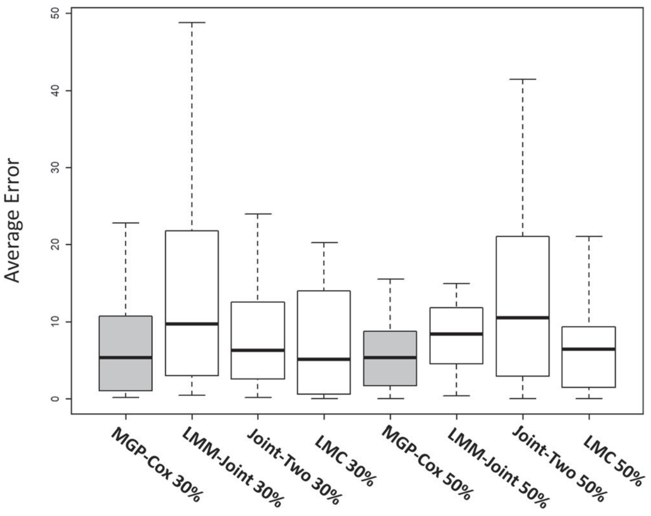

Figure 4. Remaining life prediction accuracy from Battery data.

Second, the results show that the MGCP-Cox clearly outperforms LMM-joint. This result highlights the dangers of parametric modeling and demonstrates the ability of our nonparametric approach to avoid model misspecifications.

Third, as expected, prediction errors significantly decrease as the lifetime percentiles increase. Thus, the prediction accuracy from the MGCP-Cox model will become more accurate as t^* increases and more data are collected from an online monitored unit.

Fourth, the prediction accuracy slightly decreases as we predict over a longer horizon (i.e., prediction is better for the near future). This is intuitively understandable as accuracy might decrease when predicting over a large region where not many training data might be observed. Lastly, one striking feature, shown in Figures 2–4, is that even with a small number of observations (30% observation percentile) from the testing unit we were still able to get accurate predictive results. This crucial in many applications as its allows early prediction of an event occurrence such as a disease or machine failure.

We also report the computation time in Table 1. In the table, denote by GP-joint the joint model using the ordinary GP. Denote by LMC the joint model using the LMC GP. Our model is the joint model using convolution GP. In the last column, we report the computation time of the parametric joint model LMM as a reference. Compared with the exact GP, both our model and LMC can significantly reduce the computation time, as mentioned in Section 3.2. Note that although the parametric joint model has lower complexity, it is built on the strong homogeneous assumptions and the performance decays when there is a heterogeneous pattern in the data.

In summary, the results highlight that the joint model framework can provide accurate predictions of both time series signals, event probabilities and survival time. The unique

Table 1. Computation time (in sec).

Experiment	GP-joint	Our model	LMC	LMM	GP-RNN	Deep-S	SVM
Simulation data	3615.0	60.1	78.0	3.0	70.2	44.3	4.7
NASA data	8.5 (hr)	733.2	755.0	12.3	NA	NA	NA
Battery data	9.1 (hr)	778.5	750.0	13.1	NA	NA	NA

smoothing kernel G_{ik} for each individual allows flexibility in the prediction, since it enables each training signal to have its own characteristics. This substantiates the strength of the MGCP. Equipped with the shared latent processes, the model can infer the similarities among all units, and predict signal trajectory by borrowing strength from training units. These shared and convolved latent processes in turn propagate uncertainty to the Cox model to provide a better map of both survival and time series data.

5. Conclusion

We have presented a nonparametric joint modeling framework for time series and survival data. A sparse variational inference framework is established to jointly estimate parameters from the MGCP-Cox model and propagate uncertainty from time series data to the survival model. Empirical studies highlight the advantageous features of our model compared to the state of the art methods. Our proposed model is computationally efficient and requires small number of training units. Furthermore, our framework can correctly detect failure event at early stage and is capable of avoiding some catastrophic consequences. In conclusion, our modeling framework is promising and we will consider applying it to more complex real-world applications in the future. We hope our work spurs interest in the merits of joint model.

Supplementary Materials

Appendix: This appendix includes technical details. In Appendix A, we introduce the detailed covariance functions. In Appendix B, we discuss the detail of Cox model (file type: PDF).

R code: R code for our algorithm. (zipped file)

References

Alaa, A. M., Hu, S., and van der Schaar, M. (2017), "Learning From Clinical Judgments: Semi-Markov-Modulated Marked Hawkes Processes for Risk Prognosis," in *Proceedings of the 34th International Conference on Machine Learning* (Vol. 70), JMLR.org, pp. 60–69. [2]

Alaa, A. M., and van der Schaar, M. (2017), "Deep Multi-Task Gaussian Processes for Survival Analysis With Competing Risks," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc., pp. 2326–2334. [3]

Álvarez, M., and Lawrence, N. D. (2009), "Sparse Convolved Gaussian Processes for Multi-Output Regression," in *Advances in Neural Information Processing Systems*, pp. 57–64. [4]

——— (2011), "Computationally Efficient Convolved Multiple Output Gaussian Processes," *Journal of Machine Learning Research*, 12, 1459–1500. [1,4,7]

Álvarez, M., Luengo, D., Titsias, M., and Lawrence, N. D. (2010), "Efficient Multioutput Gaussian Processes Through Variational Inducing Kernels," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 25–32. [1]

Burt, D., Rasmussen, C., and Wilk, M. V. D. (2019), "Rates of Convergence for Sparse Variational Gaussian Process Regression," arXiv no. 1903.03571. [6,7]

Cheng, C. (2018), "Multi-Scale Gaussian Process Experts for Dynamic Evolution Prediction of Complex Systems," *Expert Systems With Applications*, 99, 25–31. [1]

Cox, D. R. (1972), "Regression Models and Life-Tables," *Journal of the Royal Statistical Society*, Series B, 34, 187–202. [3]

Crowther, M. J., Abrams, K. R., and Lambert, P. C. (2012), "Flexible Parametric Joint Modelling of Longitudinal and Survival Data," *Statistics in Medicine*, 31, 4456–4471. [1]

——— (2013), "Joint Modeling of Longitudinal and Survival Data," *The Stata Journal*, 13, 165–184. [1]

Cunningham, J. P., Shenoy, K., and Sahani, M. (2008), "Fast Gaussian Process Methods for Point Process Intensity Estimation," in *Proceedings of the 25th International Conference on Machine Learning*, ACM, pp. 192–199. [3]

Damianou, A., and Lawrence, N. D. (2013), "Deep Gaussian Processes," in *Artificial Intelligence and Statistics*, pp. 207–215. [2]

Dempsey, W. H., Moreno, A., Scott, C. K., Dennis, M. L., Gustafson, D. H., Murphy, S. A., and Rehg, J. M. (2017), "iSurvive: An Interpretable, Event-Time Prediction Model for mHealth," in *International Conference on Machine Learning*, pp. 970–979. [2]

Dürichen, R., Pimentel, M. A., Clifton, L., Schweikard, A., and Clifton, D. A. (2015), "Multitask Gaussian Processes for Multivariate Physiological Time-Series Analysis," *IEEE Transactions on Biomedical Engineering*, 62, 314–322. [1]

Fernández, T., Rivera, N., and Teh, Y. W. (2016), "Gaussian Processes for Survival Analysis," in *Advances in Neural Information Processing Systems*, pp. 5021–5029. [3]

Futoma, J., Hariharan, S., and Heller, K. (2017), "Learning to Detect Sepsis With a Multitask Gaussian Process RNN Classifier," In *Proceedings of the 34th International Conference on Machine Learning* (Vol. 70), JMLR.org, pp. 1174–1182. [2,7]

Gao, R., Wang, L., Teti, R., Dornfeld, D., Kumara, S., Mori, M., and Helu, M. (2015), "Cloud-Enabled Prognosis for Manufacturing," *CIRP Annals*, 64, 749–772. [1]

Gasmi, S., Love, C. E., and Kahle, W. (2003), "A General Repair, Proportional-Hazards, Framework to Model Complex Repairable Systems," *IEEE Transactions on Reliability*, 52, 26–32. [1]

Guarnizo, C., and Álvarez, M. A. (2018), "Fast Kernel Approximations for Latent Force Models and Convolved Multiple-Output Gaussian Processes," arXiv no. 1805.07460. [1]

He, Z., Tu, W., Wang, S., Fu, H., and Yu, Z. (2015), "Simultaneous Variable Selection for Joint Models of Longitudinal and Survival Outcomes," *Biometrics*, 71, 178–187. [1]

Kalbfleisch, J. D., and Prentice, R. L. (2011), *The Statistical Analysis of Failure Time Data* (Vol. 360), New York: Wiley. [3]

Kim, M., and Pavlovic, V. (2018), "Variational Inference for Gaussian Process Models for Survival Analysis," in *Uncertainty in Artificial Intelligence*. [3]

Kontar, R., Zhou, S., Sankavaram, C., Du, X., and Zhang, Y. (2018a), "Nonparametric-Condition-Based Remaining Useful Life Prediction Incorporating External Factors," *IEEE Transactions on Reliability*, 67, 41–52. [2]

——— (2018b), "Nonparametric Modeling and Prognosis of Condition Monitoring Signals Using Multivariate Gaussian Convolution Processes," *Technometrics*, 60, 484–496. [1]

Lloyd, C., Gunter, T., Osborne, M., and Roberts, S. (2015), "Variational Inference for Gaussian Process Modulated Poisson Processes," in *International Conference on Machine Learning*, pp. 1814–1822. [3]

Mauff, K., Steyerberg, E., Kardys, I., Boersma, E., and Rizopoulos, D. (2018), "Joint Models With Multiple Longitudinal Outcomes and a Time-to-Event Outcome," arXiv no. 1808.07719. [1,7]

Mei, H., and Eisner, J. (2017), "The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process," in *Advances in Neural Information Processing Systems*, pp. 6754–6764. [3]

Moreno-Muñoz, P., Artés-Rodríguez, A., and Álvarez, M. A. (2018), "Heterogeneous Multi-Output Gaussian Process Prediction," arXiv no. 1805.07633. [1]

Pham, H. T., Yang, B., and Nguyen, T. T. (2012), "Machine Performance Degradation Assessment and Remaining Useful Life Prediction Using Proportional Hazard Model and Support Vector Machine," *Mechanical Systems and Signal Processing*, 32, 320–330. [1]

Proust-Lima, C., Séne, M., Taylor, J. M., and Jacqmin-Gadda, H. (2014), "Joint Latent Class Models for Longitudinal and Time-to-Event Data: A Review," *Statistical Methods in Medical Research*, 23, 74–90. [1]

Quiñonero-Candela, J., and Rasmussen, C. E. (2005), "A Unifying View of Sparse Approximate Gaussian Process Regression," *Journal of Machine Learning Research*, 6, 1939–1959. [4]

Ren, K., Qin, J., Zheng, L., Yang, Z., Zhang, W., Qiu, L., and Yu, Y. (2019), "Deep Recurrent Survival Analysis," in *Association for the Advancement of Artificial Intelligence*, pp. 1–8. [7]

Rizopoulos, D. (2011), "Dynamic Predictions and Prospective Accuracy in Joint Models for Longitudinal and Time-to-Event Data," *Biometrics*, 67, 819–829. [1,7]

——— (2012), *Joint Models for Longitudinal and Time-to-Event Data: With Applications in R*, Boca Raton, FL: Chapman and Hall/CRC. [1]

Rizopoulos, D., Molenberghs, G., and Lesaffre, E. M. (2017), "Dynamic Predictions With Time-Dependent Covariates in Survival Analysis Using Joint Modeling and Landmarking," *Biometrical Journal*, 59, 1261–1276. [1]

Rosenberg, P. S. (1995), "Hazard Function Estimation Using B-Splines," *Biometrics*, 51, 874–887. [6]

Ruppert, D. (2002), "Selecting the Number of Knots for Penalized Splines," *Journal of Computational and Graphical Statistics*, 11, 735–757. [6]

Saxena, A., and Goebel, K. (2008), "Turbofan Engine Degradation Simulation Data Set," Data Retrieved From NASA Ames Prognostics Data Repository, available at *https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/*. [7]

Snelson, E., and Ghahramani, Z. (2006), "Sparse Gaussian Processes Using Pseudo-Inputs," in *Advances in Neural Information Processing Systems*, pp. 1257–1264. [2,5]

Soleimani, H., Hensman, J., and Saria, S. (2018), "Scalable Joint Models for Reliable Uncertainty-Aware Event Prediction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 1948–1963. [1,2,8]

Titsias, M., and Lawrence, N. D. (2010), "Bayesian Gaussian Process Latent Variable Model," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 844–851. [1]

Tsiatis, A. A., Degruttola, V., and Wulfsohn, M. S. (1995), "Modeling the Relationship of Survival to Longitudinal Data Measured With Error. Applications to Survival and CD4 Counts in Patients With AIDS," *Journal of the American Statistical Association*, 90, 27–37. [1]

van der Wilk, M., Rasmussen, C., and Hensman, J. (2017), "Convolutional Gaussian Processes," in *Advances in Neural Information Processing Systems*, pp. 2849–2858. [2,4]

Wulfsohn, M. S., and Tsiatis, A. A. (1997), "A Joint Model for Survival and Longitudinal Data Measured With Error," *Biometrics*, 53, 330–339. [1]

Xiao, S., Yan, J., Farajtabar, M., Song, L., Yang, X., and Zha, H. (2017), "Joint Modeling of Event Sequence and Time Series With Attentional Twin Recurrent Neural Networks," arXiv no. 1703.08524. [3]

Yan, H., Liu, K., Zhang, X., and Shi, J. (2016), "Multiple Sensor Data Fusion for Degradation Modeling and Prognostics Under Multiple Operational Conditions," *IEEE Transactions on Reliability*, 65, 1416–1426. [2]

Yu, M., Law, N. J., Taylor, J. M., and Sandler, H. M. (2004), "Joint Longitudinal-Survival-Cure Models and Their Application to Prostate Cancer," *Statistica Sinica*, 14, 835–862. [1]

Yue, X., and Kontar, R. (2019a), "The Rényi Gaussian Process," arXiv no. 1910.06990. [1]

——— (2019b), "Variational Inference of Joint Models Using Multivariate Gaussian Convolution Processes," arXiv no. 1903.03867. [1]

——— (2020), "Why Non-Myopic Bayesian Optimization Is Promising and How far Should We Look-Ahead? A Study via Rollout," in *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 2808–2818. [1]

Zhao, J., and Sun, S. (2016), "Variational Dependent Multi-Output Gaussian Process Dynamical Systems," *The Journal of Machine Learning Research*, 17, 4134–4169. [1,6,7]

Zhou, Q., Son, J., Zhou, S., Mao, X., and Salman, M. (2014), "Remaining Useful Life Prediction of Individual Units Subject to Hard Failure," *IIE Transactions*, 46, 1017–1030. [1,7]

Zhu, H., Ibrahim, J. G., Chi, Y., and Tang, N. (2012), "Bayesian Influence Measures for Joint Models for Longitudinal and Survival Data," *Biometrics*, 68, 954–964. [1]