# Variational Inference for Gaussian Process Models for Survival Analysis

**Minyoung Kim***
Dept. of Electronic Engineering
Seoul Nat'l Univ. of Science & Technology
Seoul, Korea

**Vladimir Pavlovic**
Dept. of Computer Science
Rutgers University
Piscataway, NJ 08854, USA

## Abstract

Gaussian process survival analysis model (GPSAM) was recently proposed to address key deficiencies of the Cox proportional hazard model, namely the need to account for uncertainty in the hazard function modeling while, at the same time, relaxing the time-covariates factorized assumption of the Cox model. However, the existing MCMC inference algorithms for GPSAM have proven to be slow in practice. In this paper we propose novel and scalable variational inference algorithms for GPSAM that reduce the time complexity of the sampling approaches and improve scalability to large datasets. We accomplish this by employing two effective strategies in scalable GP: i) using pseudo inputs and ii) approximation via random feature expansions. In both setups, we derive the full and partial likelihood formulations, typically considered in survival analysis settings. The proposed approaches are evaluated on two clinical and a divorce-marriage benchmark datasets, where we demonstrate improvements in prediction accuracy over the existing survival analysis methods, while reducing the complexity of inference compared to the recent state-of-the-art MCMC-based algorithms.

## 1 INTRODUCTION

Survival analysis studies statistical dependencies between the time to certain event and the covariates associated with this event. This is an important problem in statistics with applications ranging from medical prognosis in clinical studies (e.g., estimating the time of cancer recurrence or remission from leukemia based on demographic and individual medical record factors), to other general areas where we seek to predict failure times of a system (e.g., bankruptcy of a firm). The key task in survival analysis is to estimate the conditional density function of the event time $t$ given the covariates $\mathbf{x}$, from which one can immediately derive several important prognostic measures. Two most common instances of such measures are the *survival function*, defined as $P(T \geq t|\mathbf{x})$, or the *prognostic index* $u(\mathbf{x})$ that quantifies the overall (anti) risk (i.e., higher index implies longer survival, and vice versa) of a patient/system with covariates $\mathbf{x}$.

A typical data-driven setting for survival analysis assumes availability of time-covariate pairs $\{(t, \mathbf{x})\}$, suggesting standard regression problem framing. However, a notable characteristic here is that some of the observed times $t$ are *censored* in the sense that we only know the actual event time is no earlier than the observed $t$. In clinical studies this typically happens when the patient exits the study or the study terminates before the event occurs[1]. Typically, the data provides the information as to whether or not each instance is censored: the event indicator variables $\delta = 1$ for event, and 0 for censored examples. Therefore, applying standard regression approaches by simply ignoring the censored samples can result in suboptimal use of data.

Several approaches have been proposed to deal with the censored instances. One way is to incorporate a cost-sensitive loss within the regression or ranking framework to learn the prognostic index function $u(\mathbf{x})$ (Shivaswamy et al., 2007; Khan & Zubek, 2008; Van Belle et al., 2009; Van Belle et al., 2011). The main idea here is to impose

*Also affiliated with Rutgers University.

[1]This type of censoring is often referred to as the *right-censoring*. *Left-censoring* applies to instances when the event time is never greater than the observed time, while the *interval-censoring* refers to the cases of observed events within an interval. Most cases in practice deal with right-censoring, which we restrict to in this paper.

an asymmetric loss on the incorrect prediction for censored examples. That is, we penalize the over-estimates (i.e., $u(\mathbf{x}) > t$) less than the under-estimates ($u(\mathbf{x}) < t$). However, these approaches focus on the prognostic index function directly and are, therefore, inherently unable to provide a measure of uncertainty, namely the distribution of the survival time $t$ for a given input $\mathbf{x}$.

The conditional density $P(t|\mathbf{x})$ is most commonly modeled using the Cox Proportional Hazard (CoxPH) model (Cox, 1972). The model represents the distribution of the survival time as a first event arrival time in a heterogeneous Poisson process. Unlike the standard Poisson process models, the intensity function (often referred to as the *hazard* function) has a dependency on the input covariates, denoted as $\lambda(t|\mathbf{x})$. The CoxPH model further factorizes the hazard function over $t$ and $\mathbf{x}$ (see (2) in Sec. 2), allowing simplicity in hazard modeling by separating the input-dependent risk factors from the time-varying effects.

Recent efforts have focused on extending the CoxPH model to address its two drawbacks: i) the proportional and non-crossing hazard rates across instances originating from the factorized form of the hazard function can be too restrictive and oftentimes unlikely, and ii) the lack of proper treatment of uncertainty in the hazard function. (Dempsey et al., 2017) extended the model by introducing latent, continuous-time Markov dynamics to address the former limitation. The latter issue can be resolved by imposing Bayesian priors on the hazard function. Although few approaches along this direction showed initial success (Hjort et al., 2010; Iorio et al., 2009; Martino et al., 2011), they either have practical limitations (e.g., how to incorporate expert knowledge) or fail to overcome the former assumption of the proportional hazard rates. Another related method is the Random Survival Forest (Ishwaran et al., 2008), which can be seen as a generalization of the Kaplan Meier method, the traditional nonparametric hazard function estimator. Recently the Deep Survival Analysis (DSA) method was proposed (Ranganath et al., 2016), which utilizes a deep hierarchical Bayesian model for survival analysis.

To address those limitations, Gaussian process survival analysis model (GPSAM) was proposed in (Fernández et al., 2016). A GP-priored latent function on the joint input space $(t, \mathbf{x})$, coupled with a non-negative link function, introduces stochasticity and removes the factorization assumption of CoxPH. The key advantage is that the Gaussian process circumvents the difficulty of modeling the hazard dependent jointly on $(t, \mathbf{x})$ through the use of the covariance (kernel) function (Rasmussen & Williams, 2006). In essence, the GPSAM supplements the proportional hazard models with additional flexibility, while being able to account for uncertainty in the hazard function.

Nevertheless, the inference in the GPSAM model is challenging because the likelihood depends on the latent function values *for an uncountable range* of time inputs $t \in \mathbb{R}_+$ (Sec. 2.2 for details), and not limited to only those induced by data in standard GP models. Motivated by the sophisticated MCMC inference algorithm for GP-priored Poisson event models (Adams et al., 2009), the authors in (Fernández et al., 2016) proposed a tractable MCMC dynamics for the GPSAM by exploiting the idea of thinning-based sampling with auxiliary variables. However, the MCMC inference algorithm often exhibits slow convergence. Despite adopting the random feature kernel approximation strategy (Rahimi & Recht, 2008) to circumvent the computationally intensive matrix inversions, the MCMC inference for GPSAM proposed in (Fernández et al., 2016) incurs considerable computational issues when applied to real applications.

In this paper, we propose two novel variational inference algorithms for GPSAM, which address the computational deficiencies of the MCMC approach. To tackle the scalability of the GP nonparametric inference, we incorporate two approximations: the pseudo-input treatment (Titsias, 2009) and the random feature expansion (Rahimi & Recht, 2008). The former approach is similar to (Lloyd et al., 2015) variational inference in the GP modulated Poisson process. However, our approach is different in that we consider the GP latent function in the joint input space within the survival analysis setup. Solutions to variational inference in both approaches admit analytic forms aside from the fast univariate Monte-Carlo estimation of expected log-likelihood. We empirically demonstrate superior performance of our proposed methods over existing survival analysis approaches on several synthetic and real benchmark datasets.

## 2 BACKGROUND

In this section we briefly review the CoxPH model with two popular parameter estimation methods. Then we discuss the recent Gaussian process survival analysis model (GPSAM) (Fernández et al., 2016) that addresses the known drawbacks of the CoxPH model.

### 2.1 COX PROPORTIONAL HAZARD MODEL

The CoxPH model (Cox, 1972; Kleinbaum & Klein, 2005) represents the conditional density

$$P(t|\mathbf{x}) = \lambda(t|\mathbf{x}) \cdot \exp\left( - \int_0^t \lambda(\tau|\mathbf{x})\, d\tau \right), \quad (1)$$

where $t \in \mathbb{R}_+$ is the time of the event (e.g., death or cancer recurrence), and $\mathbf{x} \in \mathbb{R}^d$ is the $d$-dim covariates of the subject (e.g, patient's medical features). In (1), $\lambda(t|\mathbf{x})$ is referred to as the *hazard* function, and can be interpreted as the probability of the immediate death at $t$ given that the survival time is at least $t$. The hazard function is the *intensity* function of the (inhomogeneous) Poisson process (Ross, 2006) with (1) being the first event time density, however, in survival analysis this intensity is different from subject to subject, determined by the input covariates $\mathbf{x}$.

In the CoxPH model, the hazard function is specifically assumed to follow the factorized parametric form:

$$\lambda(t|\mathbf{x}) = \lambda_0(t) \cdot \exp\left(\mathbf{b}^\top \mathbf{x}\right), \tag{2}$$

where the model parameters are comprised of the weight vector $\mathbf{b} \in \mathbb{R}^d$ and the non-negative function $\lambda_0(\cdot)$. The latter is known as the *base hazard* function which is typically modeled by the Weibull or a piecewise constant function. The consequence of the factorized form in (2) is that the hazard ratio between two subjects (with $\mathbf{x}$ and $\mathbf{x}'$) is constant over time, solely dependent on the covariates (i.e., $e^{\mathbf{b}^\top(\mathbf{x}-\mathbf{x}')}$). Also, the hazard functions of different subjects are non-crossing with each other.

Given the training data $\mathcal{D} = \{(\delta_n, t_n, \mathbf{x}_n)\}_{n=1}^N$ where $\delta_n \in \{0, 1\}$ indicates whether the observation $n$ is event ($\delta_n = 1$) or right-censored ($\delta_n = 0$), the traditional maximum likelihood learning aims to maximize the data log-likelihood $\sum_{n=1}^N \log FL(n)$ where

$$FL(n) = P(t_n|\mathbf{x}_n)^{\delta_n} \cdot P(T \geq t_n|\mathbf{x}_n)^{1-\delta_n}. \tag{3}$$

Often we name it the *full-likelihood* to differentiate it from the partial likelihood, discussed next.

Alternatively, also very popular in survival analysis, the parameters can be learned by the *partial likelihood* maximization. The notion of the partial likelihood comes from an alternative view of the data generation process. Namely, at a given time $t$, we consider a random process of selecting a subject $\mathbf{x}$ that will face an event at $t$, among all survivors at that moment. The likelihood of this is proportional to the hazard value $\lambda(t|\mathbf{x})$. More specifically, for each event instance $n$ ($\delta_n = 1$), we can regard $(t_n, \mathbf{x}_n)$ as the selected sample among the survivors $\{(t_j, \mathbf{x}_j) : t_j \geq t_n\}$, regardless of $\delta_j$'s. The so-called partial likelihood is then defined as:

$$PL(n) = \frac{\lambda(t_n|\mathbf{x}_n)}{\sum_{j:t_j \geq t_n} \lambda(t_n|\mathbf{x}_j)}, \tag{4}$$

and we maximize $\sum_{n=1}^N \delta_n \log PL(n)$.

## 2.2 GAUSSIAN PROCESS SURVIVAL MODEL

Abbreviated as GPSAM, the model aims to address the known drawbacks of the CoxPH model discussed in Sec. 1 by endowing more flexibility and accounting for uncertainty in the hazard function. This is done by imposing Gaussian process prior on the hazard function and having the latent function dependent on both $t$ and $\mathbf{x}$. More specifically,

$$\lambda(t|\mathbf{x}) = \lambda_0(t) \cdot g(f(t, \mathbf{x})), \;\; f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)). \tag{5}$$

Here $g(\cdot)$ is a non-negative link function to prevent the hazard from being negative. In (Fernández et al., 2016), they used the sigmoid $g(y) = 1/(1 + e^{-y})$, and the Weibull for the base hazard, $\lambda_0(t) = c \cdot t^{r-1}$ for $c > 0, r \geq 1$, which subsumes the constant functions ($r = 1$). Note that the kernel function operates on the joint input space $\mathbb{R}_+ \times \mathbb{R}^d$. (Fernández et al., 2016) adopted the composite kernel

$$k((t, \mathbf{x}), (t', \mathbf{x}')) = \sum_{j=1}^{d} x(j)\, x'(j)\, k_j(t, t'), \tag{6}$$

where $x(j)$ indicates the $j$-th element of $\mathbf{x}$. The kernel on time space, $k_j(t, t')$ is typically chosen as the squared exponential for $j = 1, \ldots, d$,

$$k_j(t, t') = s_j^2 \exp(-0.5(t - t')^2 / l_j^2), \tag{7}$$

with the variance and length-scale parameters $(s_j^2, l_j^2)$.

The inference in the GPSAM model is in general difficult mainly due to the form of the likelihood (1), in which the latent function $f(\cdot)$ is involved with *all* $\tau \in [0, t]$. In other words, infinitely many Gaussian latent variables need to be dealt with in principle. In (Fernández et al., 2016) they adopted the thinning-based MCMC sampling strategy motivated from (Adams et al., 2009), where the key idea is to sample[2] from the (inhomogeneous) Poisson process with intensity $\lambda_0(t)$ while keeping all the thinned samples as auxiliary state variables in the inference of the latent variables[3]. For the censored examples $n$, we can do the same thing as if $t_n$'s were exact, but remove all terms related to $t_n$ from the likelihood function.

However, the MCMC algorithm is generally slow to converge. Furthermore, each MCMC step requires the kernel matrix inversion to sample from the conditional Gaussian given both the data and the thinned samples. As the number of thinned samples can be orders of magnitude larger than the data size, they also had to resort

[2]This sampling must be easy since $\lambda_0(t) = c \cdot t^{t-1}$ admits a closed-form inverse cumulative function.

[3]Note that this thinning-based sampling is valid since they used the sigmoid link $g(\cdot)$, namely $\lambda_0(t)$ is always an upper bound of $\lambda(t|\mathbf{x})$ in (5).

to the random feature expansion trick (Rahimi & Recht, 2008) to circumvent the matrix inversion. Nevertheless, the thinned samples can grow arbitrarily large, incurring serious computational overhead. This motives our work of variational inferences in the following section.

# 3 VARIATONAL INFERENCE

We begin with the full joint model of the GPSAM with the observed data $\mathcal{D} = \{(\delta_n, t_n, \mathbf{x}_n)\}_{n=1}^N$:

$$P_\theta(\mathcal{D}, f) = P_{\theta_0}(\mathcal{D}|f) \cdot P_{\theta_k}(f). \tag{8}$$

Here $\theta = \{\theta_0, \theta_k\}$ indicates the model parameters, where $\theta_0 = (c, r)$ is the parameters of the Weibull base hazard $\lambda_0(t) = c \cdot t^{r-1}$, and $\theta_k$ denotes all kernel parameters, specifically $\{(s_j^2, l_j^2)\}$ in (7). The latter determines the Gaussian process prior $P(f)$.

The conditional data likelihood $P(\mathcal{D}|f)$ in (8) can have either of two different forms. If we follow the full likelihood (3), then the log-likelihood can be written as:

$$\log P(\mathcal{D}|f) = \sum_{n=1}^{N} \left[ \delta_n \cdot \log \lambda(t_n|\mathbf{x}_n) - \int_0^{t_n} \lambda(\tau|\mathbf{x}_n)\, d\tau \right], \tag{9}$$

with $\lambda(t|\mathbf{x})$ from (5). See Appendix A in the supplemental material for the detailed derivations. If we adopt the partial likelihood (4) instead, then $\log P(\mathcal{D}|f)$ becomes:

$$\sum_{n=1}^{N} \delta_n \cdot \left[ \log \lambda(t_n|\mathbf{x}_n) - \log \sum_{j:t_j \geq t_n} \lambda(t_n|\mathbf{x}_j) \right]. \tag{10}$$

The posterior distribution of the latent function $P_\theta(f|\mathcal{D})$ is analytically intractable, and we introduce a tractable density family $Q_\alpha(f)$ with the parameters $\alpha$, and search for $\alpha$ that makes $Q_\alpha(f)$ as close as possible to the true posterior. In defining the variational density family $Q(\cdot)$, it should be noted that we have to deal with infinitely many latent variables from $f(\cdot)$. To this end, we adopt two recent scalable variational inference algorithms: the pseudo-input approximation (Titsias, 2009) and the random feature expansion (Rahimi & Recht, 2008). We frame each of the approaches within GPSAM. They are described in the following sections.

In addition, we use the square non-negative link function, i.e., $g(y) = y^2$ instead of the sigmoid, for its merit in analytic derivation of the objective especially in Sec. 3.1, similar in nature as (Lloyd et al., 2015). That is, the hazard function given the latent function is determined as:

$$\lambda(t|\mathbf{x}) = \lambda_0(t) \cdot f(t, \mathbf{x})^2. \tag{11}$$

## 3.1 APPROXIMATION WITH PSEUDO INPUTS

To address the intractability of dealing with $f(\cdot)$ at $(\tau, \mathbf{x})$ for all time epochs $\tau$ in the domain, stemming from the likelihood (1), we first adopt the scalable pseudo-input approximation techniques recently introduced in (Titsias, 2009; Dezfouli & Bonilla, 2015; Lloyd et al., 2015). We essentially assume that there are $M$ pseudo inputs denoted by $\mathcal{Z} = \{z_1, \ldots, z_M\} \subset \mathbb{R}_+ \times \mathbb{R}^d$ (denoting $z_i = (\bar{t}_i, \bar{\mathbf{x}}_i)$), where $M$ is typically chosen to be small so that the inversion of $(M \times M)$ matrices can be done efficiently. The pseudo inputs can be thought of as the representative points for the joint input space (Quiñonero-Candela & Rasmussen, 2005). We choose the pseudo inputs by clustering the points in the pool that is formed by Cartesian product of uniformly sampled times and randomly sampled covariates from data, although they can also be learned from the data itself.

We define the variational density for the posterior as:

$$Q_\alpha(f) = \int Q_\alpha(\mathbf{f}_\mathcal{Z}) P(f|\mathbf{f}_\mathcal{Z})\, d\mathbf{f}_\mathcal{Z}. \tag{12}$$

Here we use the vector notation for the latent function: for a set $\mathcal{A} = \{(\hat{t}_i, \hat{\mathbf{x}}_i)\}_{i=1}^p \subset \mathbb{R}_+ \times \mathbb{R}$, we denote by $\mathbf{f}_\mathcal{A}$ the $p$-dim vector of the function values on the inputs $(\hat{t}_i, \hat{\mathbf{x}}_i) \in \mathcal{A}$. The central idea that enables scalability and tractability in (12) is that we only model the low-dimensional density $Q_\alpha(\mathbf{f}_\mathcal{Z})$ while all the other function values can be inferred using $P(f|\mathbf{f}_\mathcal{Z})$, the conditional density derived from the GP prior. We let $Q_\alpha(\mathbf{f}_\mathcal{Z})$ be a Gaussian with diagonal covariance, namely

$$Q_\alpha(\mathbf{f}_\mathcal{Z}) = \mathcal{N}(\mathbf{f}_\mathcal{Z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{13}$$

where $\alpha = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ with $(M \times 1)$ mean vector $\boldsymbol{\mu}$ and the $(M \times M)$ diagonal covariance matrix $\boldsymbol{\Sigma}$.

The variational parameters $\alpha$ can be found by minimizing the KL divergence between the true posterior and the variational density (12):

$$\mathrm{KL}\big(Q_\alpha(f)||P_\theta(f|\mathcal{D})\big) = \log P_\theta(\mathcal{D}) - \mathcal{L}_{PI}(\theta, \alpha), \tag{14}$$

where $\mathcal{L}_{PI}(\theta, \alpha)$ is defined as:

$$\mathcal{L}_{PI}(\theta, \alpha) = \mathbb{E}_{Q(f)}\big[\log P(\mathcal{D}|f)\big] - \mathrm{KL}\big(Q(\mathbf{f}_\mathcal{Z})||P(\mathbf{f}_\mathcal{Z})\big). \tag{15}$$

Using the non-negativity of the KL divergence in (14), the $\mathcal{L}_{PI}$ becomes a lower bound of the log-evidence,

$$\log P_\theta(\mathcal{D}) \geq \mathcal{L}_{PI}(\theta, \alpha). \tag{16}$$

Increasing $\mathcal{L}_{PI}(\theta, \alpha)$ wrt $\alpha$ renders the variational density closer to the true posterior, whereas improving it wrt the model parameters $\theta$ can *potentially*[4] improve the data

[4]Similarly as in other standard variational inferences, this does not guarantee to improve the evidence $\log P(\mathcal{D})$ since the inequality (16) is not tight.

likelihood of the model. Hence, we maximize $\mathcal{L}_{PI}(\theta, \alpha)$ wrt all parameters to achieve both variational inference and model selection simultaneously.

Next, we provide detailed derivations necessary to evaluate the objective $\mathcal{L}_{PI}$. Since the second term in the right hand side of (15) is the straightforward KL divergence between two Gaussians, we focus on the expected conditional likelihood term.

For the two forms of the likelihood, full in (9) and partial in (10), we write the expectation as:

$$\mathbb{E}_{Q(f)}\big[\log P(\mathcal{D}|f)\big] = \sum_{n=1}^{N} \big(A_n - B_n\big), \tag{17}$$

where $A_n$ is the expectation of the log-hazard at data point $n$,

$$A_n = \delta_n \cdot \Big(\log \lambda_0(t_n) + \mathbb{E}_{Q(f_n(t_n))}\big[\log f_n(t_n)^2\big]\Big), \tag{18}$$

with the abbreviation $f_n(t) = f(t, \mathbf{x}_n)$. Whereas $A_n$ is shared by both likelihoods, $B_n$ is defined differently.

$$B_n^f = \int_0^{t_n} \lambda_0(\tau) \cdot \mathbb{E}_{Q(f_n(\tau))}\big[f_n(\tau)^2\big] d\tau, \tag{19}$$

is for the full likelihood, while the following is for the partial likelihood:

$$B_n^p = \delta_n \cdot \mathbb{E}_{Q(f)}\left[\log \sum_{j:t_j \geq t_n} \lambda_0(t_n) \cdot f_j(t_n)^2\right]. \tag{20}$$

**Full likelihood**. Note that in (18) and (19) the distributions over which the expectations are taken are univariate Gaussians, specifically from (12), $Q(f_n(t)) = \mathcal{N}(\tilde{\mu}_n(t), \tilde{\sigma}_n^2(t))$ where

$$\tilde{\mu}_n(t) = k_{(t,\mathbf{x}_n),\mathcal{Z}} \mathbf{K}^{-1} \mu, \tag{21}$$

$$\tilde{\sigma}_n^2(t) = k_{(t,\mathbf{x}_n),(t,\mathbf{x}_n)} - k_{(t,\mathbf{x}_n),\mathcal{Z}} \mathbf{K}^{-1} k_{\mathcal{Z},(t,\mathbf{x}_n)} + k_{(t,\mathbf{x}_n),\mathcal{Z}} \mathbf{K}^{-1} \Sigma \mathbf{K}^{-1} \; k_{\mathcal{Z},(t,\mathbf{x}_n)}. \tag{22}$$

For two time-covariates input sets $\mathcal{A}_1$ and $\mathcal{A}_2$, we denote by $k_{\mathcal{A}_1,\mathcal{A}_2}$ the $(|\mathcal{A}_1| \times |\mathcal{A}_2|)$ kernel matrix obtained by applying $k(\cdot,\cdot)$ on $(\mathcal{A}_1 \times \mathcal{A}_2)$. Also, $\mathbf{K} = k_{\mathcal{Z},\mathcal{Z}}$ indicates the $(M \times M)$ kernel matrix on the pseudo inputs $\mathcal{Z}$. The expectation of the squared-log term in (18) can be done by the Monte-Carlo estimation. For it is *univariate* sampling, this does not incur any significant computational overhead. As we also need to compute gradients of $A_n$ wrt the parameters of the sampling distribution $Q(f_n(t_n))$, we adopt the re-parametrized Gaussian sampling technique (Kingma & Welling, 2014) to reduce the variance of the estimate. More specifically, we express the random samples from $Q(f_n(t_n))$, denoted by $f_n^{(s)}(t_n)$ for $s = 1, \ldots, S$, as:

$$f_n^{(s)}(t_n) = \tilde{\mu}_n(t_n) + (\tilde{\sigma}_n^2(t_n))^{1/2} \epsilon_n^{(s)}, \tag{23}$$

where $\epsilon_n^{(s)} \sim \mathcal{N}(0,1)$. The expectation in $A_n$ is then estimated as:

$$\frac{1}{S} \sum_{s=1}^{S} \log\Big(\tilde{\mu}_n(t_n) + (\tilde{\sigma}_n^2(t_n))^{1/2} \epsilon_n^{(s)}\Big)^2. \tag{24}$$

We sample $\{\epsilon_n^{(s)}\}$ once, and fix them throughout the optimization, which empirically performed better with a lower variance than re-sampled at each iteration. As we separate randomness ($\epsilon_n^{(s)}$) from the parameters to be optimized, the gradient of (24) can be computed straightforwardly while yielding an unbiased estimate of the gradient of the original (18). Optionally, we can further reduce the variance of the estimate by using the Rao-Blackwellization technique (Casella & Robert, 1996).

For $B_n^f$ in (19), due to the square link function, the inner expectation equals $\tilde{\mu}_n(\tau)^2 + \tilde{\sigma}_n^2(\tau)$, which allows the integral to be derived analytically for the composite squared exponential kernel (6) and (7) (c.f. (Lloyd et al., 2015)). However, the analytic derivation has to resort to certain confluent hyper-geometric function which can be numerically unstable. In the experiments, we rather adopt the numerical integration by having uniformly sampled grid points over the time horizon.

**Partial likelihood**. Looking into $B_n^p$ in (20), the key difference from the above full likelihood derivation stems from the multiple latent variables (i.e., $\{f_j(t_n)\}_{j:t_j \geq t_n}$) that are dependent on one another, preventing us from taking advantage of the univariate sampling. Since the number of these variables (i.e., $|\{j : t_j \geq t_n\}|$) can be as large as the entire data set, naively sampling from $Q(f)$ jointly can yield an estimate with large variance. Instead, we consider the upper bound of $B_n^p$ (leading to a lower bound on $\mathcal{L}_{PI}$) using the Jensen's inequality. Specifically, from (20),

$$B_n^p \leq \delta_n \cdot \log \sum_{j:t_j \geq t_n} \lambda_0(t_n) \cdot \mathbb{E}_{Q(f_j(t_n))}\Big[f_j(t_n)^2\Big]. \tag{25}$$

The square link allows an analytical expression of the expectation in $\tilde{B}_n^p$, $\tilde{\mu}_j(t_n)^2 + \tilde{\sigma}_j^2(t_n)$, which can be evaluated easily using (21) and (22), likewise its gradients. Note that (regardless of whether we use the upper bound or not) the base hazard term $\log \lambda_0(t_n)$ cancels out with that in (18) in the final objective (17), preventing one from learning $\lambda_0(t)$. This is an inherent problem originating in the hazard form (5), where $\lambda_0(t)$ is shared across examples. To this end, we simply borrow the estimate of $\lambda_0(t)$ from the full-likelihood learning.

### 3.2 RANDOM FEATURES APPROXIMATION

To deal with uncountably many random variables and their matrix inversions brought about from the nonparametric Bayesian Poisson process GPSAM, we consider the random features expansion as an alternative approximation strategy. The central idea of the random features (Rahimi & Recht, 2008; Cho & Saul, 2009) is to seek a finite dimensional feature vector representation for input such that the inner product on this feature space equals (in expectation) the kernel value. For the composite kernel function (6), its GP-priored latent function $f(t, \mathbf{x})$ can be approximated by:

$$\hat{f}(t, \mathbf{x}) = \frac{1}{m} \sum_{j=1}^{d} x(j) \Big( \mathbf{a}_j^\top \cos(\boldsymbol{\omega}_j t) + \mathbf{b}_j^\top \sin(\boldsymbol{\omega}_j t) \Big), \tag{26}$$

where $\mathbf{a}_j$, $\mathbf{b}_j$, and $\boldsymbol{\omega}_j$ are all $m$-dim iid random variables (samples) with $\mathbf{a}_j, \mathbf{b}_j \sim \mathcal{N}(0, s_j^2 \mathbf{I}_m)$ and $\boldsymbol{\omega}_j \sim \mathcal{N}(0, \frac{1}{l_j^2} \mathbf{I}_m)$ for $j = 1, \ldots, d$. Here $\mathbf{I}_m$ denotes the $(m \times m)$ identity matrix. We let $\mathbf{a} = \{\mathbf{a}_j\}_{j=1}^d$ and the others similarly. It can be shown that $\mathrm{Cov}(\hat{f}(t, \mathbf{x}), \hat{f}(t', \mathbf{x}')) = k((t, \mathbf{x}), (t', \mathbf{x}'))$, which implies that the finite dimensional random variables $\{\mathbf{a}, \mathbf{b}, \boldsymbol{\omega}\}$ are sufficient to represent the latent function $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$. The parameter $m$ is the number of random samples to approximate the kernel, which trades off: large $m$ reduces the approximation error at the cost of computational overhead.

In this treatment, we aim to infer the posterior distribution $P(\mathbf{a}, \mathbf{b}, \boldsymbol{\omega} | \mathcal{D})$, and the variational inference reduces to maximizing:

$$\mathcal{L}_{RF}(\theta, \beta) = \mathbb{E}_{Q(\mathbf{a}, \mathbf{b}, \boldsymbol{\omega})} \big[ \log P(\mathcal{D} | \hat{f}) \big] - \mathrm{KL}(\theta, \beta), \tag{27}$$

where we use the fully factorized variational density, $Q(\mathbf{a}, \mathbf{b}, \boldsymbol{\omega}) = Q(\mathbf{a}) Q(\mathbf{b}) Q(\boldsymbol{\omega})$, each modeled as a Gaussian, $Q(\mathbf{a}) = \prod_{j=1}^d \mathcal{N}(\mathbf{a}_j; \boldsymbol{\mu}_j^a, \boldsymbol{\Sigma}_j^a)$ and similarly for the others. Here $\boldsymbol{\mu}_j^a$ and $\boldsymbol{\Sigma}_j^a$ are $(m \times 1)$ mean vector and $(m \times m)$ diagonal covariance matrix, respectively, and $\beta$ includes all these variational parameters. The term $\mathrm{KL}(\theta, \beta)$ denotes the sum of individual KL terms for $\mathbf{a}$, $\mathbf{b}$, and $\boldsymbol{\omega}$; for instance, for $\mathbf{a}$, we have: $\mathrm{KL}(Q(\mathbf{a}) || \mathcal{N}(\mathbf{a}; 0, s_j^2 \mathbf{I}_m))$. As before, these KL terms all involve Gaussians, having analytic forms, easy to evaluate and take derivatives.

Similarly to Sec. 3.1, the expected log-likelihood term in (27) has two variations, full or partial likelihood, and we decompose it into two parts exactly the same way as (17). For concreteness, with the abbreviation $\hat{f}_n(t) = \hat{f}(t, \mathbf{x}_n)$, we approximate the expected log-likelihood term in (27) using the re-parametrized Monte-Carlo estimate. That is, after sampling $\boldsymbol{\epsilon}_j^{a(s)}$, $\boldsymbol{\epsilon}_j^{b(s)}$ and $\boldsymbol{\epsilon}_j^{\omega(s)}$ independently from $\mathcal{N}(0, \mathbf{I}_m)$ for $j = 1, \ldots, d$ and $s = 1, \ldots, S$, the sampled version of the random weight vector $\mathbf{a}_j$ is formed as ($\mathbf{b}_j^{(s)}$ and $\boldsymbol{\omega}_j^{(s)}$ similarly):

$$\mathbf{a}_j^{(s)} = \boldsymbol{\mu}_j^a + (\boldsymbol{\Sigma}_j^a)^{1/2} \boldsymbol{\epsilon}_j^{a(s)}, \tag{28}$$

Then the posterior samples $\hat{f}_n^{(s)}(t)$ can be written as:

$$\sum_{j=1}^{d} \frac{x_n(j)}{m} \Big( \big(\mathbf{a}_j^{(s)}\big)^\top \cos(\boldsymbol{\omega}_j^{(s)} t) + \big(\mathbf{b}_j^{(s)}\big)^\top \sin(\boldsymbol{\omega}_j^{(s)} t) \Big), \tag{29}$$

With these sampled functions, one can basically obtain the estimate of the objective (27) that can be decomposed into forms similar to (18), (19), and (20), which we denote by $\hat{A}_n$, $\hat{B}_n^f$, and $\hat{B}_n^p$, respectively. Although evaluating $\hat{A}_n$ and $\hat{B}_n^p$ (and their gradients) can be done similarly as in Sec. 3.1 with no additional difficulty, working on $\hat{B}_n^f$ introduces a new computational challenge. Unlike the pseudo-input approximation where we were able to express $\mathbb{E}_Q \big[ f_n(\tau)^2 \big]$ as an analytic form $\tilde{\mu}_n(\tau)^2 + \tilde{\sigma}_n^2(\tau)$, we can only estimate the expectation numerically using the samples (29) from $Q(\cdot)$. However, we have an outer integration of this expectation over $\tau \in [0, t_n]$, and the (grid-based) numerical integration would incur computational explosion as we need $(d \cdot m \cdot S \cdot G)$ samples/numbers involved in, where $G$ is the number of grid points over $[0, t_n]$ (typically, $S$ and $G$ are several thousands, and $d \approx 10$).

To address this difficulty, we propose an alternative estimation strategy for $\hat{B}_n^f$. We regard $\lambda_0(\tau) = c \cdot \tau^{r-1}$ in the integration as an unnormalized density, more specifically, $\lambda_0(\tau) = \rho_n \cdot p_n(\tau)$ where $p_n(\tau)$ is the density having the support $[0, t_n]$ and $\rho_n = \int_0^{t_n} \lambda_0(\tau) d\tau = \frac{c}{r} t_n^r$ is the normalizer. Thus $p_n(\tau) = r \frac{\tau^{r-1}}{t_n^r}$ over $[0, t_n]$. Then

$$\hat{B}_n^f = \rho_n \cdot \mathbb{E}_{p_n(\tau)} \Big[ \mathbb{E}_Q \big[ f_n(\tau)^2 \big] \Big]. \tag{30}$$

This allows us to sample $\tau_n^{(s)} \sim p_n(\tau)$ for $s = 1, \ldots, S$ independently with the random features/weights in (28), and estimate $\hat{B}_n^f$ as:

$$\rho_n \cdot \frac{1}{S} \sum_{s=1}^{S} \hat{f}_n^{(s)}(\tau^{(s)})^2. \tag{31}$$

Note that sampling from $p_n(\tau)$ can be done using the inverse transform sampling: for its CDF is $F_n(\tau) = (\tau / t_n)^r$, the samples $\tau_n^{(s)}$ can be expressed as:

$$\tau_n^{(s)} = F_n^{-1}(u^{(s)}) = t_n \cdot (u^{(s)})^{1/r}, \tag{32}$$

where $u^{(s)}$ are uniform samples from $[0, 1]$. We further reduce the variance of the estimate by using the same re-parametrization trick, namely plugging (32) into (31) to separate the randomness from the parameters.

# 4 EMPIRICAL EVALUATIONS

The performance of the proposed variational inference methods is demonstrated on both synthetic and real datasets. Our approaches are denoted as follows: $\mathbf{VI}^f_{PI}$ and $\mathbf{VI}^p_{PI}$ are the approximations based on pseudo inputs in Sec. 3.1 with full and partial likelihood, respectively, while $\mathbf{VI}^f_{RF}$ and $\mathbf{VI}^p_{RF}$ indicate the variational inference with random feature expansions described in Sec. 3.2. The competing approaches are summarized, with abbreviations, as:

- **MCMC**: The MCMC-based inference method for GPSAM (Fernández et al., 2016), where we used hyperparameters similar to theirs.
- **CoxPH**$^f$ and **CoxPH**$^p$: The full and partial likelihood maximization learning of the CoxPH model.
- **SVCR**: The support vector censored regression approach (Shivaswamy et al., 2007) with no cost imposed for the over-estimation of the censored samples.
- **SVRC**: Another regression-based approach (Khan & Zubek, 2008) that adopts asymmetric costs for over-estimation depending on the violation types.
- **MINLIP**: The ranking-based prognostic function estimation method (Van Belle et al., 2009), which enforces the correct ordering of the prognostic indices for time-comparable pairs of samples. The method further aims to preserve the relative time differences in the prognostic indices.
- **Model2**: The hybrid approach that attempts to combine the ranking constraints and the cost-sensitive loss in estimating the prognostic function (Van Belle et al., 2011).

For the performance measures, we focus on the accuracy of the estimated prognostic index function $u(\mathbf{x})$. Whereas the approaches based on regression and/or ranking directly estimate $u(\mathbf{x})$, for the CoxPH-based methods we derive it naturally by $u(\mathbf{x}) = -\mathbf{b}^\top \mathbf{x}$ from the learned CoxPH models. For GPSAM, where the hazard function is not factorized, we estimate the expected event time $\mathbb{E}[t|\mathbf{x}]$ as the survival index. Two performance measures popular in survival analysis are considered: i) `Concordance index` – the proportion of the pairs of samples whose predicted survival times are correctly ordered (i.e., $(u(\mathbf{x}_i) - u(\mathbf{x}_j))(t_i - t_j) \geq 0$), and ii) `Log-rank-`$\chi^2$ statistics – the statistical test score measuring the difference between two risk groups formed by thresholding the prognostic indices by their median. For both measures, higher scores are better.

For our variational inference methods, we vary the following hyperparameters: the number of pseudo inputs ($M$) for the $\text{VI}_{PI}$ and the number of random features ($m$) for the $\text{VI}_{RF}$. We use the best set obtained by cross validation on the held-out portion of the training data, where the concordance index is used as the selection criterion. The optimal parameters are, consistently across all datasets, $M = 20$ for the $\text{VI}_{PI}$ and $m = 50$ for the $\text{VI}_{RF}$. In the latter part of the section and in the supplemental material (Appendix B), we also compare the performances and running times of the other parameter settings. The number of samples for the Monte-Carlo estimation in our variational inference is fixed as 3000. The MCMC approach for GPSAM model (Fernández et al., 2016) used $m = 50$ random features, and the number of MCMC iterations is chosen as 5000 with the first 1000 samples dropped out. The hyperparameters of the other competing models (e.g., the regularization parameters in the regressions) are also determined by cross validation.

## 4.1 SYNTHETIC DATASETS

To judge the effectiveness and flexibility of the proposed variational inference methods for GPSAM, we consider a synthetic scenario where the data samples are generated from a non-proportional hazard model. In particular, we consider a stratified CoxPH model, which can be seen as a conditional mixture of several CoxPH models. More specifically, the hazard function is defined as $\lambda(t|\mathbf{x}) = \lambda_{s(\mathbf{x})}(t) \cdot \exp(\mathbf{b}^\top_{s(\mathbf{x})}\mathbf{x})$ where $s(\mathbf{x}) \in \{1, \ldots, K\}$ is a gating function among $K$ component CoxPH models. The base hazard function $\lambda_s(t)$ for each component model $s = 1, \ldots, K$, is defined to be the Weibull function $c_s \cdot t^{r_s - 1}$ with different parameters $(c_s, r_s)$ for each $s$. The gating function follows a piecewise linear form, $s(\mathbf{x}) = \arg\max_{1 \leq s \leq K} \mathbf{w}_s^\top \mathbf{x}$, where we choose $\mathbf{w}_s$'s randomly. We set the number of base models as $K = 3$. The input covariates $\mathbf{x}$ are sampled randomly from $\mathcal{N}(0, \mathbf{I})$.

To mimic the censoring process, for each generated sample, we randomly turn it into a censored one with probability $p$. The observed time $t$ for the censored sample is then uniformly sampled from $[0, t_*]$ where $t_*$ is the original value before censoring. We choose $p = 0.3$. After generating 100 samples, we perform 10-fold cross validation where the averaged test scores with standard deviations are depicted in Table 1. For each measure, the best performing method in terms of the average value is boldfaced. To measure the statistical significance, we also conducted the Wilcoxon signed-rank tests, pairwisely against the (boldfaced) best performing method. With the null hypothesis that two approaches result in statistically indistinguishable performance, the $p$-values

Table 1: (Synthetic dataset) Average test prediction performance. Our variational approximation approaches, $\text{VI}_{PI}$ and $\text{VI}_{RF}$, adopt $M = 20$ pseudo inputs and $m = 50$ random features, respectively. Best average score method is boldfaced. Parentheses indicate the $p$-values from the Wilcoxon signed rank test against the best (boldfaced) approaches.

| Methods | C-Index (%) | Log-Rank-$\chi^2$ |
|---|---|---|
| $\text{VI}^f_{PI}$ | $\mathbf{83.04 \pm 1.91}$ $(--)$ | $\mathbf{13.88 \pm 2.85}$ $(--)$ |
| $\text{VI}^p_{PI}$ | $79.68 \pm 2.50$ $(0.002)$ | $10.71 \pm 4.73$ $(0.125)$ |
| $\text{VI}^f_{RF}$ | $81.55 \pm 2.20$ $(0.049)$ | $10.55 \pm 5.66$ $(0.250)$ |
| $\text{VI}^p_{RF}$ | $81.28 \pm 2.54$ $(0.049)$ | $10.73 \pm 5.38$ $(0.250)$ |
| MCMC | $77.20 \pm 3.66$ $(0.002)$ | $10.81 \pm 6.00$ $(0.193)$ |
| $\text{CoxPH}^f$ | $73.43 \pm 5.51$ $(0.002)$ | $8.70 \pm 2.88$ $(0.232)$ |
| $\text{CoxPH}^p$ | $73.27 \pm 4.99$ $(0.002)$ | $9.14 \pm 3.37$ $(0.160)$ |
| SVCR | $68.32 \pm 6.51$ $(0.002)$ | $5.55 \pm 3.48$ $(0.027)$ |
| SVRC | $73.05 \pm 4.37$ $(0.002)$ | $7.81 \pm 3.20$ $(0.084)$ |
| MINLIP | $65.75 \pm 4.34$ $(0.002)$ | $3.40 \pm 2.20$ $(0.004)$ |
| Model2 | $71.32 \pm 3.04$ $(0.002)$ | $5.63 \pm 2.24$ $(0.027)$ |

are shown in the tables.

The proposed variational inference approaches, both $\text{VI}_{PI}$ and $\text{VI}_{RF}$, exhibit superior generalization performance compared to all contrasted methods. Specifically, the pseudo-input approximation optimizing the full-likelihood ($\text{VI}^f_{PI}$) performs the best in both measures. With regard to the concordance index, it is significantly better than all other models ($p$-values $< 0.05$), but leads to marginal improvements in the log-rank-$\chi^2$ measure. The CoxPH-based models ($\text{CoxPH}^f$ and $\text{CoxPH}^p$) are mostly outperformed by the GPSAM models due to the substantial mismatch with the true data process (simplified modeling assumption of non-crossing hazard functions across instances). Compared to the MCMC approach (Fernández et al., 2016), our proposed VI methods yield higher prediction accuracies, related to improved hazard function estimation. In contrast, the MCMC potentially suffers from computational overhead, preventing convergence to the target distribution.

To investigate the computational benefits of the proposed approaches over the MCMC algorithm, we measure the actual inference times (for our variational inference, we record the entire running time until convergence). Implemented in MATLAB and run on 2.4GHz Intel Xeon CPU, the running times are: $571.7$ seconds for $\text{VI}^f_{PI}$, $1165.1$ seconds for $\text{VI}^f_{RF}$, and $8121.2$ seconds for the MCMC, indicating significant computational advantage of our VI approaches. The log-likelihood scores $\log P(\mathcal{D}_*)$ of GPSAM evaluated on the test data $\mathcal{D}_*$ are also summarized in Table 3. Although the scores are mostly comparable, the MCMC attains the highest likelihood. However, it should be noted that we only compute lower bounds of the log-likelihoods (i.e., $\mathcal{L}_{PI}$ in (15) and $\mathcal{L}_{Rf}$ in (27)) for our variational learning.

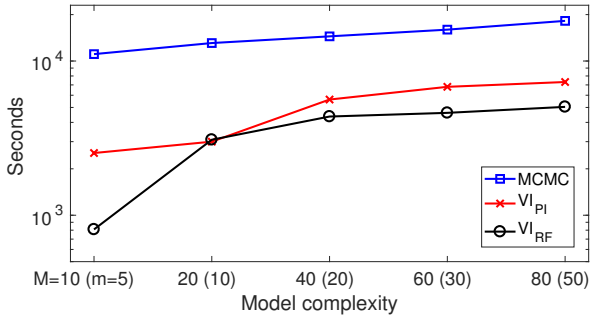


Figure 1: The inference times of three methods: MCMC, $\text{VI}^f_{PI}$, and $\text{VI}^f_{RF}$ on the MLC dataset.

## 4.2 REAL DATASETS

Next we test the performance of the proposed methods on two clinical and one non-clinical datasets: i) (`VLC`) Veteran's lung cancer dataset (Prentice, 1974; Kalbfleisch & Prentice, 2002) contains records of 137 patients with 6 covariates such as age, weight, treatment, cell type, and disease history, ii) (`MLC`) Mayo lung cancer dataset (Therneau & Grambsch, 2000) having 167 patients with similar 7 covariates, and iii) (`Divorce`) dataset (Lillard & Panis, 2000) which records the divorce years since marriage for 3771 couples, among which we use a fifth of the samples randomly chosen. The proportions of the censored samples are: $7\%$ for VLC, $28\%$ for MLC, and $70\%$ for Divorce, where the study involved in the latter dataset often fails to track the status of many of the couples, which leads to the large proportion of the censored samples. The Divorce dataset originally provides three categorical features for each sample: the husband education years (categorized into three levels of less than 12 yrs, more than 15, and between two), whether the husband is African American or not, and whether the ethnicity of the couple is different or not. We also introduce additional nonlinear features of pairwise and triple products, yielding 7-dim covariates. Other experimental settings follow the synthetic experiment.

We performed 10-fold cross validation where the test results are shown in Table 2. The best methods, boldfaced with $p$-values from the statistical test against other methods, are mostly our variational inference methods (four out of six). Our approaches estimate the posterior of the latent function more accurately than the MCMC while enjoying the benefits of the GP to account for uncertainty in the hazard modeling and relaxing the time-covariates factorized assumption of the CoxPH model.

Table 2: (Real datasets) Average test prediction performance. Our variational approximation approaches, $\text{VI}_{PI}$ and $\text{VI}_{RF}$, adopt the number of pseudo inputs $M = 20$ and random features $m = 50$, respectively. Best method in terms of the average value is boldfaced. Figures in parentheses indicate the $p$-values from the Wilcoxon signed rank test against the best (boldfaced) approaches.

| Datasets | VLC | | MLC | | Divorce | |
|---|---|---|---|---|---|---|
| Methods | C-Index (%) | Log-Rank-$\chi^2$ | C-Index (%) | Log-Rank-$\chi^2$ | C-Index (%) | Log-Rank-$\chi^2$ |
| $\text{VI}^f_{PI}$ | $71.99 \pm 3.67$ $(0.0020)$ | $3.89 \pm 2.74$ $(0.3750)$ | $\mathbf{72.68 \pm 3.00}$ $(--)$ | $2.74 \pm 2.07$ $(0.6250)$ | $63.60 \pm 3.17$ $(1.0000)$ | $2.27 \pm 1.70$ $(1.0000)$ |
| $\text{VI}^p_{PI}$ | $70.87 \pm 4.75$ $(0.0039)$ | $3.61 \pm 2.82$ $(0.2754)$ | $69.44 \pm 6.48$ $(0.0273)$ | $2.35 \pm 2.27$ $(0.4316)$ | $62.89 \pm 3.92$ $(0.1934)$ | $2.25 \pm 1.97$ $(0.7695)$ |
| $\text{VI}^f_{RF}$ | $\mathbf{76.79 \pm 4.08}$ $(--)$ | $5.13 \pm 4.17$ $(1.0000)$ | $68.08 \pm 4.98$ $(0.0098)$ | $1.64 \pm 1.94$ $(0.1934)$ | $63.73 \pm 2.67$ $(0.0703)$ | $2.07 \pm 1.50$ $(0.7695)$ |
| $\text{VI}^p_{RF}$ | $76.49 \pm 4.51$ $(0.6875)$ | $\mathbf{5.81 \pm 4.27}$ $(--)$ | $67.00 \pm 5.35$ $(0.0098)$ | $1.32 \pm 1.92$ $(0.1309)$ | $\mathbf{64.56 \pm 3.47}$ $(--)$ | $2.24 \pm 1.33$ $(1.0000)$ |
| MCMC | $68.48 \pm 2.57$ $(0.0098)$ | $2.34 \pm 1.54$ $(0.0840)$ | $66.46 \pm 6.38$ $(0.0039)$ | $2.15 \pm 2.72$ $(0.4316)$ | $63.81 \pm 3.15$ $(1.0000)$ | $\mathbf{2.35 \pm 2.68}$ $(--)$ |
| $\text{CoxPH}^f$ | $69.28 \pm 6.05$ $(0.0098)$ | $3.12 \pm 3.75$ $(0.0547)$ | $66.89 \pm 11.05$ $(0.3223)$ | $3.25 \pm 2.53$ $(1.0000)$ | $63.33 \pm 2.71$ $(0.4922)$ | $1.67 \pm 1.38$ $(0.5566)$ |
| $\text{CoxPH}^p$ | $69.10 \pm 5.88$ $(0.0098)$ | $2.64 \pm 3.86$ $(0.0078)$ | $68.25 \pm 11.14$ $(0.1934)$ | $\mathbf{3.65 \pm 3.04}$ $(--)$ | $63.33 \pm 2.71$ $(0.4922)$ | $1.67 \pm 1.38$ $(0.5566)$ |
| SVCR | $55.41 \pm 9.31$ $(0.0020)$ | $1.10 \pm 1.68$ $(0.0020)$ | $56.42 \pm 17.89$ $(0.0039)$ | $2.52 \pm 2.71$ $(0.1602)$ | $54.38 \pm 11.21$ $(0.0098)$ | $0.87 \pm 0.72$ $(0.3750)$ |
| SVRC | $55.36 \pm 9.66$ $(0.0020)$ | $1.10 \pm 1.68$ $(0.0020)$ | $54.93 \pm 17.65$ $(0.0039)$ | $2.61 \pm 2.74$ $(0.2324)$ | $54.43 \pm 10.88$ $(0.0059)$ | $1.14 \pm 1.19$ $(0.3223)$ |
| MINLIP | $65.64 \pm 9.79$ $(0.0098)$ | $2.59 \pm 3.23$ $(0.1309)$ | $68.60 \pm 9.28$ $(0.2754)$ | $3.18 \pm 3.03$ $(0.8457)$ | $51.75 \pm 5.38$ $(0.0039)$ | $0.60 \pm 0.96$ $(0.1602)$ |
| Model2 | $57.40 \pm 10.82$ $(0.0039)$ | $1.59 \pm 2.07$ $(0.0020)$ | $68.81 \pm 8.06$ $(0.1934)$ | $3.19 \pm 3.02$ $(0.6953)$ | $55.83 \pm 10.45$ $(0.0195)$ | $0.69 \pm 0.74$ $(0.2324)$ |

Next we compare running times. To see the effect of the approximation model complexity on the inference time, we vary the parameters in model learning. Specifically, $M$ is chosen from $\{10, 20, 40, 60, 80\}$ for $\text{VI}_{PI}$ and $m$ is from $\{5, 10, 20, 30, 50\}$ for $\text{VI}^f_{RF}$ and also the MCMC method (Fernández et al., 2016) that employs the random feature approximation. All methods are implemented in MATLAB run on 2.4GHz Intel Xeon CPU. The results on the MLC dataset are visualized in Fig. 1. Results demonstrate that our variational methods are an order of magnitude faster than the MCMC while achieving comparable or often superior prediction performance. For other datasets, refer to Appendix B in the supplemental material. Finally, the test log-likelihood scores of the attained models are depicted in Table 3. Considering we report lower VI bounds of the log-likelihoods, all proposed methods exhibit comparable, or superior, generalization performance to that of the MCMC.

Table 3: Average test log-likelihood scores attained by $\text{VI}_{PI}$ ($M = 20$), $\text{VI}_{RF}$ ($m = 50$), and the MCMC.

| Datasets | Synthetic | VLC | MLC | Divorce |
|---|---|---|---|---|
| $\text{VI}^f_{PI}$ | $-36.40$ | $-31.50$ | $-40.00$ | $-80.31$ |
| $\text{VI}^p_{PI}$ | $-41.13$ | $-35.00$ | $-41.78$ | $-84.10$ |
| $\text{VI}^f_{RF}$ | $-35.18$ | $-25.87$ | $-45.48$ | $-86.75$ |
| $\text{VI}^p_{RF}$ | $-36.33$ | $-26.05$ | $-45.47$ | $-86.75$ |
| MCMC | $-34.01$ | $-30.70$ | $-38.61$ | $-105.53$ |

## 5 CONCLUSION

We have proposed a family of novel and highly scalable variational inference methods for the Gaussian process survival analysis model. Our approaches can estimate the posterior of the GP latent function in this flexible non-proportional hazard model more accurately, with running times an order of magnitude faster than the state-of-the-art MCMC algorithm.


### Acknowledgments

MK is supported by National Research Foundation of Korea (NRF-2016R1A1A1A05921948).

## References


Adams, R. P., Murray, I., & MacKay, D. J. (2009). Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. International Conference on Machine Learning.

Casella, G., & Robert, C. P. (1996). Rao-Blackwellisation of sampling schemes. *Biometrika*, *83*, 81–94.

Cho, Y., & Saul, L. K. (2009). Kernel methods for deep learning. In Advances in Neural Information Processing Systems.

Cox, D. (1972). Regression models and life-tables (with discussion). *Journal of the Royal Statistical Society, Series B*, *34*, 187–220.

Dempsey, W. H., Moreno, A., Scott, C. K., Dennis, M. L., Gustafson, D. H., Murphy, S. A., & Rehg, J. M. (2017). iSurvive: An Interpretable, Event-time Prediction Model for mHealth. International Conference on Machine Learning.

Dezfouli, A., & Bonilla, E. V. (2015). Scalable inference for Gaussian process models with black-box likelihoods. In Advances in Neural Information Processing Systems.

Fernández, T., Rivera, N., & Teh, Y. W. (2016). Gaussian processes for survival analysis. In Advances in Neural Information Processing Systems.

Hjort, N. L., Holmes, C., Mller, P., & Walker, S. G. (2010). *Bayesian nonparametrics*. Cambridge University Press.

Iorio, M. D., Johnson, W. O., Mller, P., & Rosner, G. L. (2009). Bayesian nonparametric nonproportional hazards survival modeling. *Biometrics*, *65*, 762–771.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. *The annals of applied statistics*, *2*, 841–860.

Kalbfleisch, J., & Prentice, R. (2002). *The statistical analysis of failure time data*. Wiley Series in Probability and Statistics, New York.

Khan, F., & Zubek, V. (2008). Support vector regression for censored data (SVRc): A novel tool for survival analysis. In Proceedings of the Eighth IEEE International Conference on Data Mining (ICDM).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In Proceedings of the Second International Conference on Learning Representations.

Kleinbaum, D. G., & Klein, M. (2005). *Survival analysis: A self-learning text (statistics for biology and health)*. Springer.

Lillard, & Panis (2000). aml multilevel multiprocess statistical software. Release 1.0, EconWare, LA, California.

Lloyd, C., Gunter, T., Osborne, M. A., & Roberts, S. J. (2015). Variational inference for Gaussian process modulated Poisson processes. International Conference on Machine Learning.

Martino, S., Akerkar, R., & Rue, H. (2011). Approximate Bayesian inference for survival models. *Scandinavian Journal of Statistics*, *38*, 514–528.

Prentice, R. L. (1974). A log gamma model and its maximum likelihood estimation. *Biometrika*, *61*, 539–544.

Quiñonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, *6*, 1939–1959.

Rahimi, A., & Recht, B. (2008). Random features for large-scale kernel machines. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), Advances in Neural Information Processing Systems 20.

Ranganath, R., Perotte, A., Elhadad, N., & Blei, D. (2016). Deep survival analysis. Machine Learning for Health Care.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. The MIT Press.

Ross, S. M. (2006). *Simulation*. Academic Press.

Shivaswamy, P., Chu, W., & Jansche, M. (2007). A support vector approach to censored targets. In Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM).

Therneau, T. M., & Grambsch, P. M. (2000). *Modeling survival data: Extending the Cox model*. Springer-Verlag, New York.

Titsias, M. K. (2009). Variational learning of inducing variables in sparse Gaussian processes. In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics.

Van Belle, V., Pelckmans, K., Suykens, J., & Van Huffel, S. (2009). Learning transformation models for ranking and survival analysis. Tech. Rep., 09-45, ESAT-SISTA, K.U.Leuven (Leuven, Belgium).

Van Belle, V., Pelckmans, K., Van Huffel, S., & Suykens, J. (2011). Support vector methods for survival analysis: A comparison between ranking and regression approaches. *Artificial Intelligence in Medicine*, *53*, 107–118.