# Bayesian Transformation Models for Multivariate Survival Analysis with Applications in Large Data

By

David Jeffrey Schlueter

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Biostatistics

December 15, 2018

Nashville, Tennessee

Approved:

Robert E. Johnson, Ph.D.

Qingxia Chen, Ph.D.

Christopher Fonnesbeck, Ph.D.

Daniel Fabbri, Ph.D.

# ACKNOWLEDGEMENTS

During the course of my studies at Vanderbilt, I have met so many individuals who have profoundly impacted my growth as a professional statistician. First and foremost, I must sincerely thank my dissertation advisor Dr. Cindy Chen, for guiding the way and teaching me how to perform research rigorously. Without her brilliance, patience, and persistence, I would not have been able to complete this dissertation. Secondly, I would like to thank the individuals who served on my committee for their excellent insight and suggestions for the development of this dissertation. Drs. Daniel Fabbri, Christopher Fonnesbeck, and Robert Johnson, I am eternally indebted to you for your contribution to this project.

My studies at Vanderbilt would not have been possible without financial support from my research assistantship. I would like to thank Kathy Gracey at the Vanderbilt Center of Excellence for Children in State Custody for the collaborative opportunity and especially Dr. Rameela Chandrasekhar, my RA advisor. Working with Rameela has had a profound impact on how I conduct my statistical collaborations.

Next I would like to thank the Vanderbilt University graduate program, especially Dr. Jeffrey Blume and Amanda Harding for their immense help during the course of my studies here.

Finally, I would like to thank my friends and family who have supported me throughout this entire process. A huge thank you to Mom, Dad, Liz, Erica, and my partner Serena.

## TABLE OF CONTENTS

# LIST OF TABLES

## LIST OF FIGURES

Figure Page

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ABPM | Average Bias of Posterior Mean |
| CDF | Cumulative Distribution Function |
| CP | Coverage Probability of Posterior Credible Interval |
| HMC | Hamiltonian Monte Carlo |
| MCMC | Markov Chain Monte Carlo |
| MSE | Mean Squared Error |
| NUTS | No-U-Turn Sampling |
| SDPM | Standard Deviation of Posterior Mean |
| SDS | Standard Deviation of Simulations |
| VI | Variational Inference |

# CHAPTER 1

# INTRODUCTION

## 1.1 Basics of Survival Analysis

At its core, this dissertation concerns itself with the study of survival analysis. For proper understanding of the the contents within, it is first necessary to remind the reader of several basic concepts and quantities commonly found in survival analytic statistical procedures.

Survival analysis is principally concerned with the modeling of *time-to-event data*, i.e., the time until individuals in a population experience a particular event of interest (Klein and Moeschberger, 2005). Time-to-event data are especially prevalent in the field of biostatistics since many health studies are concerned with following patients over time with respect to some health outcome. For example, in a trial assessing the efficacy of a drug targeted at preventing an adverse outcome such as myocardial infarction (MI), an individual's time until experiencing MI could be collected prospectively as part of the study. Alternatively, in the case of an observational study, time until an event can be derived in a variety of ways including using time-to-admittance to a hospital, for example. Additionally, after controlling for other covariates (e.g. age and sex), our goal is to be able to model the time to myocardial infarction as a function of a treatment or some other independent variable of interest. In this example, one would interpret longer time periods of not experiencing the event (in other words, survival with respect to MI) as more desirable. Hence, if individuals in a hypothetical treatment arm of the study survive longer (in a statistically meaningful manner) without experiencing myocardial infarction, we would conclude that the treatment is efficacious.

A natural question, however, is determining the methodology by which we intend to *model* such time-to-event data. Naturally, as is the case in any statistical framework, we seek to model the data with a mathematical procedure that intrinsically accounts for the variability in the data set. Of course, this implies a probabilistic formulation of data modeling wherein observed data are treated as observations of a positive-valued continuous random variable, say, $T$. One resulting question, however, is which probabilistic statements are appropriate to model these time-to-event data? In other words, which functions related to the probability distribution associated with $T$ are most appropriate to model so that we can use the data to make inference about how $T$ is influenced by the treatment variable (or other predictors of interest)? If we return to our preceding myocardial infarction example, our question of interest might be phrased: *do patients in the treatment arm go longer without experiencing an MI than those in the control group?* Alternatively, stated, one could ask *for a given time t, what is the probability that a person does not experience MI past t.* In mathematical notation, if we consider an individual's time as a random quantity, $T$, we would be interested in developing a model for $S(t) = P(T > t)$. Specifying a probabilistic quantity like $S(t)$ as a function of covariates, however, is challenging. Instead of modeling the survival probability function directly, it is often more mathematically and computationally convenient to model related quantities, which we will discuss now.

The most commonly modeled alternative to the survival function is the *hazard function* (Klein and Moeschberger, 2005). The *hazard* is defined as the instantaneous risk of failure (or experience of the event of interest) and is mathematically given by

$$\lambda(t) = \lim_{\delta t \to 0} \frac{P(t < T \leq t + \delta t | T > t)}{\delta t}. \tag{1.1}$$

For a specified distribution function $F(t) = P(T \leq t)$, the *probability density* *function* is given by

$$\begin{aligned} f(t) &= \frac{\partial F(t)}{\partial t} \\ &= \lambda(t)S(t). \end{aligned}$$

One last important function in this context known as the *cumulative hazard* which is defined as

$$\Lambda(t) = \int_0^t \lambda(x)dx \tag{1.2}$$

and is related to the aforementioned survival function by

$$S(t) = \exp(-\Lambda(t)). \tag{1.3}$$

In principle, the specification of a mathematical model that can model the survival probability can be developed by modeling any of the three aforementioned quantities directly. Some of these functions, however, are more easily modeled than others. One of the most common strategies for modeling survival data is to introduce a covariate effect on either the hazard or cumulative hazard. Specifically, the *proportional hazards* model assumes that the hazard between two values of an independent variable differs by some proportional amount. Mathematically, for such a model, a hazard function is modeled as a function of a covariate by

$$\lambda(t|X) = \lambda_0(t)\exp(\beta^T X) \tag{1.4}$$

where $\lambda_0(t)$ is known as the baseline hazard, corresponding to the hazard when all covariates equal zero (this quantity is often treated as a nuisance variable). Alternative specifications of a survival model can be given in the form of accelerated failure time models, however these will not be discussed in this work.

### 1.1.1 Censoring

When following individuals over time, it is possible that we fail to observe the time of the event of interest during the course of the study period. Observations that fall into this category are known as *censored* observations: a concept is quite commonly observed in time-to-event analyses. There are several types of censoring, but in this dissertation we will be concerned with so-called *right-censored* models. In these models, it is assumed that censored observations have survived past the observed or recorded time with respect to particular event.

This brings us to the formalized version of the likelihood contribution for each censored subject. For a univariate model, For each individual, our observed data consists of an observation time $t_i$ and an event indicator $\delta_i$ indicating whether the event occurred. In terms of random variables, we assume that the true event time $T_i^*$ and a censoring time $C_i$ are independent and that our observed time $T_i = \min(T_i^*, C_i)$. In terms of modeling censored observations, the likelihood contribution is given by the survival function, $S(t)$; i.e. for a right-censored observation, we assume that the subject survived past the observed time $t$.

### 1.1.2 Moving beyond proportional hazards

While the proportional hazards specification of the hazard model is convenient for fitting and interpretation of coefficients, the proportionality assumption on the hazard can be quite restrictive and lead to biased results when applied to datasets containing data that violates the assumption. To illustrate, consider a scenario wherein we try to fit a univariate proportional hazards model to data simulated from non-proportional hazards (the true survival function and cumulative hazard functions are given by the green dashed line). We see that the fitted proportional hazards model produces a biased estimate of the underlying functions. With increasing data, this bias is not overcome; instead, the variability around the bias lessens and we become more confident in the wrong inference.

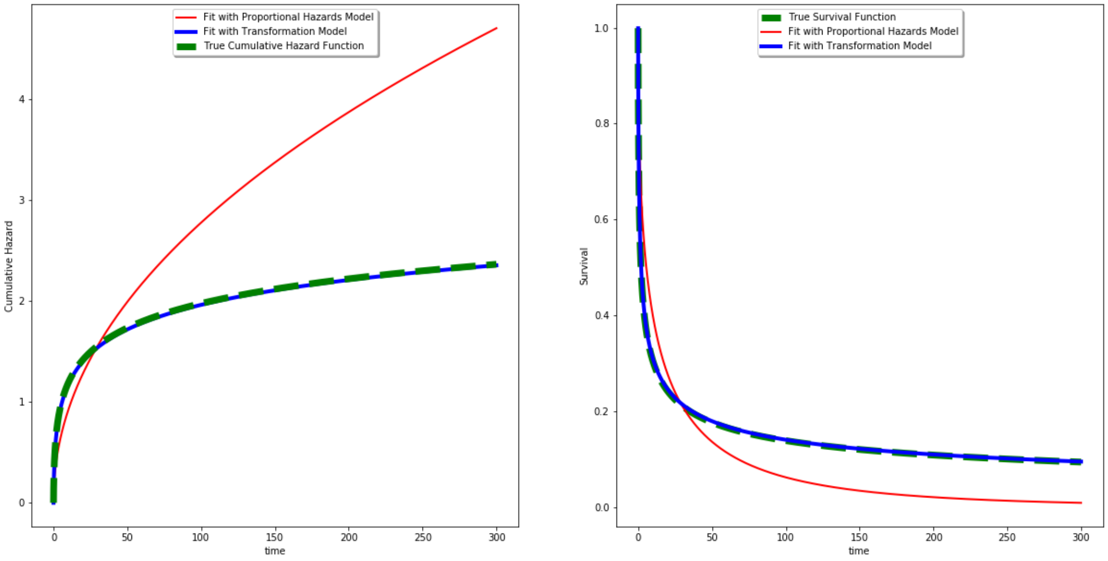


Figure 1.1: Model Fits On Non-Proportional Hazards data

Naturally, other methods may be applied to account for non-proportional hazards such as the accelerated-failure time model (Wei, 1992) and the time-varying covariate methodology (Sueyoshi, 1992). However, this dissertation primarily centers around the study and application of *generalized transformation models.* These models *generalize* common models used in survival analysis, including the aforementioned proportional hazards model and the the proportional odds model as special cases. This generalization is made through the introduction of a particular transformation function defined as

$$G(v, r) = vI(r = 0) + [\log(1 + vr)/r]I(r > 0) \tag{1.5}$$

for $v, r \in \mathcal{R}^+$ (Zeng and Lin (2007), Zeng et al. (2009)). Intuitively, as values of $r$ increase, the outputs of the transformation for the varying parameter become closer for successive values of $v$.

In this model, it is most convenient, as we will see, to model the conditional cumulative hazard instead of the hazard. Using the previously developed survival quantities, we can determine the form of the univariate likelihood function for right censored data. In the univariate case, beginning from the cumulative hazard function, we have that

$$\begin{aligned}\Lambda(t|X) &= G\left[\Lambda_0(t)\exp(\beta^T X), r\right] \\ &= \log\left[1 + r\Lambda_0(t)\exp(\beta^T X)\right]/r.\end{aligned}$$

From this specification, we conclude that the survival function and hazard functions are given respectively by

$$S(t|X) = \exp\left\{-\log\left[1 + r\Lambda(t)\exp(\beta^T X)\right]/r\right\} \tag{1.6}$$

and

$$\begin{aligned}\lambda(t|X) &= \frac{\partial\Lambda(t|X)}{\partial t} \\ &= \frac{\exp(\beta^T X)}{1 + r\Lambda(t)\exp(\beta^T X)}\frac{\partial\Lambda_0(t)}{\partial t} \\ &= \lambda(t)\exp(\beta^T X)\left\{1 + r\Lambda_0(t)\exp(\beta^T X)\right\}^{-1} \qquad (1.7)\end{aligned}$$

In this dissertation, we focus on the development of multivariate models that utilize this particular transformation model. Model parameters will be estimated using Bayesian methods, which we now briefly discuss.

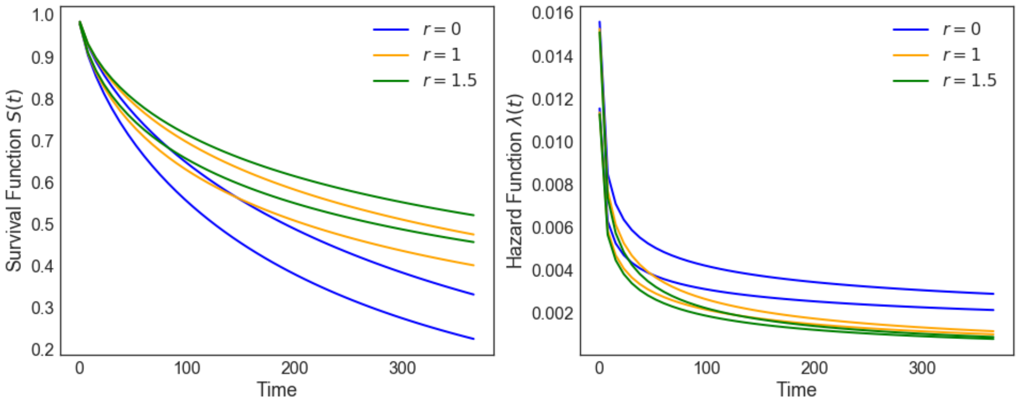


Figure 1.2: Hazard and Survival Functions subject to the Transformation $G(v, r) = vI(r = 0) + [\log(1 + vr)/r]I(r > 0)$.

## 1.2 Bayesian Statistics

### 1.2.1 Bayesian Analysis

We now consider the statistical inferential framework upon which this dissertation is founded. Bayesian inference is motivated by the idea that we can update our *prior belief* about a phenomenon by performing an experiment (Gelman et al., 2013). Operationally, we convolve the results of that experiment with our prior belief, and subsequently quantify the resulting posterior belief in some fashion (through appropriate summary statistics of this posterior distribution, for example). This process mimics the inductive accumulation of scientific knowledge, but depending on the strength of our prior belief, one particular experiment might not necessarily overwhelm our prior belief about the scientific process in which we are interested. That is to say, there exists a particular weighting between prior belief and experimentation involved in the Bayesian framework (the exact weighting depends on the experimental conditions, sample size, and strength of prior). This process of learning is predicated on a somewhat controversial notion of probability, but its implementation is a direct application of a well known probabilistic result: Bayes' theorem. The Bayesian view interprets a probability as a personal belief of an event, in contrast to the frequentist view of a long run phenomenon. Assuming a subjective, belief-based interpretation of probability, Bayes' theorem gives a recipe for updating our prior belief about a scientific process (in other words, the parameters). In the Bayesian view, the parameters of a statistical probability model represent the truth about the underlying data generating mechanism (i.e. which specific formulation of our assumed probability model gave rise to the observed data). Bayes' Theorem (Casella and Berger, 2002) gives the following formulation for learning about model parameters given our data:

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{\int P(X|\theta)P(\theta)d\theta}. \tag{1.8}$$

This theorem arises from foundational relationships between conditional and joint probabilities. One implication of using Bayes' formula as a method of learning about a parameter is that we can no longer believe that this parameter is fixed and constant (as opposed to the frequentist, long run probabilistic interpretation). Rather, Bayes' theorem implies existence of uncertainty about the parameter of interest and thus model parameters must be treated as random variables themselves.

To operationalize the process of Bayesian modeling, in principle, one specifies a probability generating model for the data encoded in the likelihood $P(X|\theta)$. To complete the formulation, one must select a prior probability distribution for the parameter $\theta$, denoted $P(\theta)$. Integrating the joint likelihood results in the marginal likelihood which comprises the denominator of the computation in equation 1.8. Although in principle, the process of Bayesian computation is a matter of leveraging an elementary probability theorem, in practice, computation of the posterior distribution can be quite complex.

### 1.2.2 Introduction to the Bayesian Fitting

The process of implementing Bayesian models began first with the restriction of models under consideration to that of the class of ‘so-called’ conjugate models. Mathematically speaking, the posterior distribution of parameters under this particular framework were the same distributional form as the prior distribution, but with updated parameters according to the conditional update. For example, for a parameter that is learned about and is endowed with a Gaussian distribution and whose generating model is assumed to be Gaussian as well, the posterior distribution of this model is also Gaussian. There are many examples of conjugate models and this strategy had proven useful for relatively simple model specifications (see Gelman et al. (2013) for more details).

Although Bayesian computation had been founded on distributional equivalencies of prior to posterior densities, the paradigm was plagued by the rigid necessity of conjugacy in its modeling. The requirement of a mathematically closed-form solutions as a means for posterior inference severely limited Bayesian models. For instance, censored and truncated models lack closed form conjugate models. This posed a serious problem for the adoption of Bayesian methods for more complex models until the introduction of simulation based-computational strategies which are still in mainstream use (Metropolis et al., 1953). The authors described a method for simulating draws from a posterior density via the construction of a Markov chain with random walk. There has been much work in the advancement of Markov Chain Monte Carlo (MCMC) sampling; see Brooks et al. (2011) for an overview.

## 1.3 Bayesian Survival Analysis

The Bayesian implementation of basic survival analysis is well founded and methodological development has been historically performed in Gibbs sampling-based software such as JAGS or OpenBugs (Ibrahim et al., 2005). Formulating such Gibbs-sampling procedures often relied on the formulation and conversion of likelihood into alternative forms. One example of this is the utilization of the so-called *Poisson trick*, in which a semi-parametric Cox proportional hazards model can be converted into a counting process-based model that is suitable for use in the software BUGS (Spiegelhalter et al., 2007). Specifically, time until death is model using the following formulation. We model the number of failures at time $t$ for a subject $i = 1, \dots\ n$ through a counting process $N_i(t)$. The counting process has associated with it an corresponding intensity process that dictates the time specific rate of failure. This in intensity process is given by

$$I_i(t) = Y_i(t)\lambda_0(t)\exp(\beta^{'}X_i)$$

where $Y_i(t) \in \{0, 1\}$ depending on whether the subject is observed at the given time point. Finally, the model is fit using a Poisson likelihood using a Gibbs sampler.

Many subsequent Bayesian survival analysis methodologies have been founded upon this formulation. However, with the introduction of tensor-based computing libraries such as 'theano' (Theano Development Team, 2016) and 'tensorflow' (Abadi et al., 2016), gradient calculations can be leveraged for the implementation of Hybrid Monte Carlo and Variational inference; as is the case in the python library 'pymc3' (Salvatier J, 2016). Tensor-based computation allows Bayesian modelers to utilize a deep-learning platform to compute otherwise computational difficult tasks (such as differentiation) and allows the modeler to not rely solely on pre-existing Gibbs sampling techniques or to be forced to build their own for every specific new model. Instead, the model likelihood can be passed as a simple tensor function within a model context.

### 1.3.1 Overview and Scope of Dissertation

This dissertation broadly focuses on the advancement and extension of Bayesian methods for multivariate survival analysis with applications in larger data settings. Increased sample information allows researchers to develop more flexible Bayesian multivariate time-to-event models, but at the cost of increased computational time associated with Markov Chain Monte Carlo (MCMC) sampling. Therefore, an additional challenge addressed in this dissertation is the testing and implementation of scalable algorithms that allow our models to be applied to large data.

In the first portion of this dissertation, we focus on generalizing various bivariate copula survival models that allow for separate specification of marginal and association components. We extend the Bayesian formulation of these models to include generalized forms of the marginal survival functions as well as to include flexible B-spline formulations of the baseline hazard. To demonstrate the approach, we apply the techniques to a myocardial infarction dataset.

In the second portion of this thesis, we develop a scalable Bayesian framework to accommodate time to first event of multivariate survival outcomes with ordinal severity. This is done using a flexible Bayesian multivariate frailty model that derestricts the form of the survival function in order to simultaneously study the correlated covariate effects on differing severity levels of the outcome and to provide a mechanism for combining these profiles into an overall effect. This work was motivated by and applied to the Tennessee Asthma Bronchiolitis Study (TABS) cohort (derived from the TennCare medical claims database) in order to quantify maternal smoking effect across levels of first hospital admittance for infant bronchiolitis. Using an additional data source correlating the multivariate survival outcomes with ordinal severity scores, we provide a systematic and flexible way to determine the overall direction of the smoking effect size over the multivariate survival events. Using Bayesian methods at the scale of an insurance claims database is challenging due to the computational bottlenecks of typical MCMC routines, which motivated the use of approximate Bayesian inference in this paper. Variational Bayesian inference is one such approximate approach, wherein the posterior density of a model parameter is approximated by a member of a distributional family closest to the true posterior in terms of some statistical distance. Additionally, it is unclear how valid these approximations are outside of relatively simple statistical models. Therefore, as a secondary focus of the paper, we study the efficacy of variational Bayesian methods on multivariate survival models through a variety of simulations.

An important contribution to the advancement of scientific research is the development of usable computational tools for researchers to easily adapt to their own projects. Therefore, as a final component to this dissertation, we provide software for the previously described time-to-event methods using the python Bayesian module ‘pymc3’ which is built upon the deep learning library ‘theano’ (Theano Development Team, 2016). This software includes flexible implementations of the methods developed in this dissertation with user-friendly syntax to reduce the barrier-of-entry to researchers. Leveraging pymc3 allows for usage of the deep learning library to automatically compute otherwise difficult quantities such as gradient information necessary for Bayesian methods as well as Hybrid Monte Carlo sampling and Variational inference. The software incorporates other pymc3 mechanics like the ‘variational’ sub-module with the intent that researchers can use these models to answer scientific questions involving multivariate time-to-event outcomes with EHR-scale datasets.

# CHAPTER 2

# BAYESIAN INFERENCE FOR MARGINAL TRANSFORMATION MULTIVARIATE SURVIVAL MODELS

## 2.1 Introduction

One pervasive problem in the modeling of multivariate failure times is the proper specification of a flexible form of joint survivor function. The frailty model, which will be discussed in a future section of this dissertation, offers a conditional version of the survival function. In this case, the survival function is specified such that individual multivariate outcomes are conditionally independent given a subject-specific random effect (relating the different dimensions of the multivariate outcome) endowed with some parametric distribution. Alternatively, one may choose to to specify the joint survival function in such a way that the marginal distributions are separated from the association structure. A *copula* function provides a means for defining a joint survival function by first specifying the marginal models and subsequently joining those models under a particular association structure.

Copula models have been studied in the context of survival analysis as early as Oakes (1989) wherein statistical inference was developed based on maximum-likelihood estimation procedures. Subsequently, fitting procedures were introduced in Shih and Louis (1995) for the association parameter in survival models. These papers provide a basis for the frequentist inferential treatment of general multivariate models in the marginal regression setting. However, much of the research in the area of survival copula methodology include an assumption of a proportional change via covariates in the marginal distributions. One notable exception to this was provided by Li et al. (2017), who studied the bivariate survival using quantile regression and martingale-based non-parametric estimation.

In practice, the proportionality assumption, whether it is about the odds or the hazard, may be too strenuous and fitting such models to data that do not follow proportionality assumptions will lead to biased estimation procedures that cannot be remedied with the introduction of larger sample sizes. As a result, there has been much work in the area of generalizing survival functions to accommodate a wider class of relationships of hazard functions for differing levels of covariates. Transformation models for survival data first appeared in the literature with Dabrowska and Doksum (1988) and later Zeng and Lin (2007) wherein the authors considered a model with cure fraction. Shortly after the development of these models, Zeng et al. (2009) developed an extension of the transformation methodology to model multiple outcomes within a conditional survival model by constructing the survival function conditionally on a random effect distributed as gamma distribution with unit mean. Here, the appropriate transformation was learned from the data in large samples or, in smaller samples, was fixed over a grid of values and model fit was evaluated based on some information criteria. One early work that generalized random effects survival analysis in the Cox proportional hazards setting to one without restriction on the form of the baseline hazard can be found in Sargent (1998). Bayesian extensions to these models can be found in both Yin (2008) and de Castro et al. (2014). A robust extension to the linear transformation model was provided by Lin et al. (2017) using rank-based estimation procedures.

Frequentist inference for marginal transformation models also has an established literature and alternative transformations models than those considered in this paper have been considered. First, Chen (2010) provides a maximum-likelihood-based copula marginal methods for semi-parametric formulations for competing risks and dependent censoring; the latter of these provides a methodology for transformed marginals using counting processes. In Li et al. (2008), the authors consider transformations of non-parametric hazard models into standard normal marginal distributions and subsequent coupling with a Gaussian copula. Similarly, Lin et al. (2014) considers probit-style transformations in order to recover a Gaussian copula model structure. A two-staged estimation methodology is provided by Chen and Yu (2012), where the marginal parameters are first estimated using martingale methods and subsequently plugged into the full likelihood for association parameter estimation. Furthermore, Diao and Yin (2012), developed a random effects cure model generalizing the proportional hazards cure model and mixture cure models using empirical processes. Finally, a non-copula based marginal transformation methodology based on estimating equations was developed in Chen and Lu (2012).

The Bayesian treatment of copula models also has a rich literature. Bayesian treatment of survival copula models with specified marginal distributions, however, was first studied in Romeo et al. (2006) wherein the authors give both one and two-stage estimation procedures using proportional hazards and non-parametric estimation of the marginal models. As an extension, Meyer and Romeo (2015) provided a Bayesian semi-parametric analysis of recurrent failure time data using copulas. The framework allowed for parametric as well as a nonparametric modeling of the marginal baseline hazards and models the influence of covariates on the marginals via a proportional hazards assumption. Another extension to this work was given by E Shemyakin and Youn (2006) wherein the authors provide a model for joint last survival in the Bayesian framework. A mixed outcome model was considered in Craiu and Sabeti (2012) wherein the authors proposed a method for jointly modeling continuous and discrete outcomes. In Louzada et al. (2013), the authors consider a Bayesian model using the Farlie-Gumbel-Morgenstern copula but do not consider a transformation in the margins. Recently, Romeo et al. (2018), explored Bayesian bivariate survival analysis using the power variance function copula. All of these papers comprise a healthy and formidable basis literature, however, to our knowledge, there has been no work addressing Bayesian copula models that flexibly define the form of the marginal survival models via transformation models of the sort developed in Zeng et al. (2009). This paper seeks to fill this void and effectively provides marginal model extension to the methods developed in de Castro et al. (2014) and a generalization of the proportional hazards assumption assumed in the parametric model of Romeo et al. (2006). Additionally, we provide an alternative semi-parametric formulation to those listed above by studying the Bayesian application of cubic splines in the baseline hazard.

In this paper, we develop generalized copula models that extend pre-existing Bayesian marginal models that require the introduction of covariates through the specification of proportional hazards models to a general transformation class that includes both the proportional hazards and proportional odds as special cases. We will use Bayesian inference to both determine the choice of transformation as well as determine the best copula model based on Bayesian information criteria. We consider two separate specifications of the marginal survival functions: first, we consider models with parametric specifications of baseline hazard, and secondly we consider models using cubic spline specifications of the baseline hazard. For the latter of these models, we study both penalized and un-penalized approaches to the spline specification.

The remainder of this paper is organized in the following manner. Section 2 introduces the rigorous definition of copula models as well as a description of the families considered in this work. Furthermore, we describe the semi-parametric formulation of the transformation copulas using splines. In this part, we first define the spline approach to modeling baseline hazard and subsequently discuss the various types of specific spline models we will be implementing. Next, we describe how the spline model is formulated for the univariate transformation case and how to extend to the copula model. Section 3 describes the inferential strategies necessary to implement the proposed models. Next, a simulation study is performed to demonstrate statistical efficacy of the proposed approach. Finally, the paper concludes with an application of the proposed method to a myocardial data set wherein risk factors for stroke and myocardial infarction are jointly estimated. We provide simulations for each of the proposed implementations and subsequently apply the methodologies to the Atherosclerosis Risk in Communities Study (ARIC) dataset. We close the entire paper with a discussion and future research.

### 2.2 Basic Quantities

In this section we present the basic definition and intuition behind copula models. An in-depth discussion may be found in Nelsen (1999) and Joe (1997). First, consider a vector of $K$ random variables $(T_1, \ldots, T_K)'$ with continuous margins (with corresponding marginal cumulative distribution functions, $F_k(t_k) = P(T_k \leq k)$, $k = 1, \ldots, K$). By the probability integral transform, the corresponding vector of CDFs has uniformly distributed marginals. Now, we define the **copula** (Nelsen, 2007) of the vector $(T_1, \ldots, T_K)^T$ as the joint CDF of the margins such that

$$C_\alpha(u_1, \ldots, u_k) = P(T_1 \leq F_1^{-1}(u_1), \ldots, T_k \leq F_K^{-1}(u_K)) \tag{2.1}$$

The function $C_\alpha : [0,1]^K \rightarrow [0,1]$ is a copula if

1. $C_\alpha(u_1, 0, \ldots, u_K) = 0$, i.e. is equal to zero if one of its components equals zero
2. $C_\alpha(1, \ldots, u, 1) = u$, i.e. the copula is equal to the marginal $u$ if all other arguments are equal to 1.
3. $C_\alpha$ is non-decreasing.

From this definition, we can begin to develop a modeling strategy for coupling disparate marginals via the following theorem.

**Sklar's theorem**: For random variables $(T_1, \ldots, T_K)^T$ with joint cumulative distribution function

$$F(t_1, t_2, \ldots, t_K) = P(T_1 \leq t_1, \ldots, T_K \leq t_K) \tag{2.2}$$

and marginals $F_k(t) = P(T_k \leq t_k)$, there exists a copula, $C_\alpha$ such that

$$F(t_1, t_2, \ldots, t_K) = C_\alpha(F_1(t_1), \ldots, F_K(t_K)). \tag{2.3}$$

Additionally, given a copula $C_\alpha$ and marginal distributions $F_k(t_k) = P(T_k \leq t_k)$, then

$$C_\alpha(F_1(t_1), \ldots, F_K(t_K)) \tag{2.4}$$

defines a valid joint cumulative distribution function.

From a modeling perspective, Sklar's Theorem allows us to separate the modeling of the marginal distributions $F_k(t_k)$ from the dependence structure. In other words, we are guaranteed the existence of a copula that gives a valid distribution function for any specification of marginal distributions. Now, instead of associating two distribution functions with a copula to define a joint CDF, we can analogously define a joint survival function in terms of a copula function (Nelsen, 2007), (Georges et al., 2001).

In this paper, we extend the formulation of bivariate copula models to include a marginal transformation that allows for a more flexible formation of the margins in the survival function thereby overcoming the restriction of proportional hazards in such models. We now proceed to discuss the classes of copula functions that will be considered throughout this paper.

### 2.2.1 Classes of Copula Functions

There exist a multitude of valid copula functions, each developed in an effort to model different underlying dependence phenomena. For a nearly complete collection of these functions, see (Nadarajah et al., 2017). The most basic implementations of copula function, however, are the parametric families which include the Gaussian and the Student's t-copula. These functions exploit the distributional assumptions of each family to properly couple the marginal distributions. The usage of a Gaussian copula for survival data was addressed by (Masarotto et al., 2012).

Although the parametric copula framework offers a tractable and simple implementation of copulas, we instead focus on the Archimedean class of copula models (Genest and Mackay, 1986). Definitionally, we say that a copula is *Archimedean* if its joint survival function has the form

$$C_\alpha(u_1, \ldots, u_K) = \psi^{-1}\left(\sum_{k=1}^{K} \psi(u_k); \alpha\right) \tag{2.5}$$

for some $\psi : [0, \infty] \to [0, 1]$. Notice that this family is parameterized by $\alpha$. This parameter effectively measures the strength of the dependence between the marginal survival models. The members of this family that will be considered in the parametric portion of this paper include the Clayton, Frank, Gumbel, and Joe Models. In this section we only present the form of the copula; relevant derivatives for likelihood formulation are available in Appendix A.

### 2.2.2 Clayton Model

The first copula under consideration was first developed by Clayton as a multivariate extension of the proportional hazards model (Clayton and Cuzick, 1985). The Laplace transform (Widder, 2015) of this particular model is that of a gamma distribution, relating this model to the gamma frailty model. Further discussion on the differences between the two models can be found in (Goethals et al., 2008)

The form of the copula function for this case is given by

$$C_\alpha(u_1, u_2) = (u_1^{-\alpha} + u_2^{-\alpha} - 1)^{-1/\alpha}. \tag{2.6}$$

An illustration of the association structure can be found in Figure 2.1.

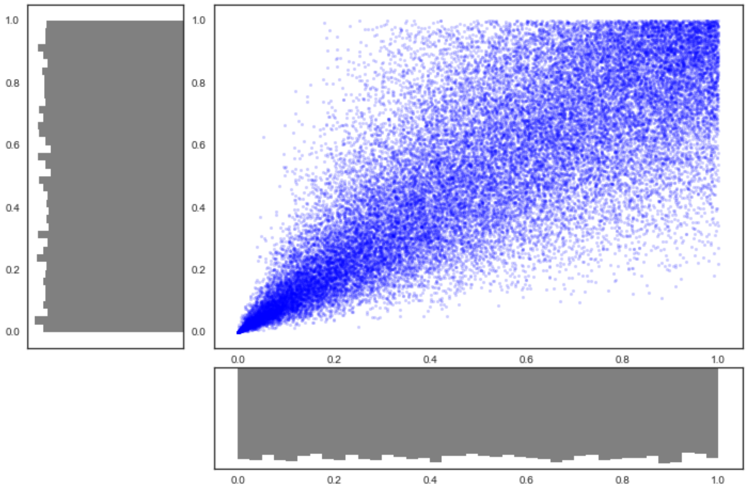


Figure 2.1: Joint Distribution of Uniform Marginals under Clayton Copula with $\alpha = 8$

### 2.2.3 Gumbel-Hougaard Model

The Gumbel-Hougaard copula is given by

$$C_\alpha(u_1, u_2) = e^{-((-\log(u_2))^\alpha - \log^\alpha(u_1))^{\frac{1}{\alpha}}}. \tag{2.7}$$

An illustration of the association structure can be found in Figure 2.2.

### 2.2.4 Joe Model

The copula specification for the Joe copula is given by:

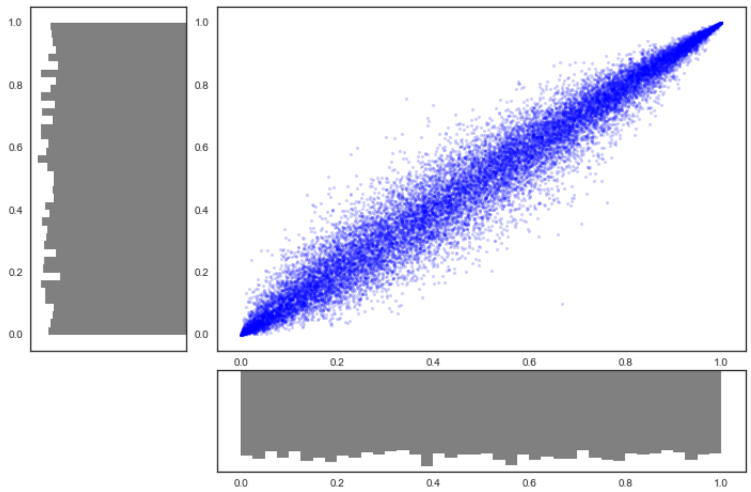


Figure 2.2: Copula Generation for Gumbel Model

$$\begin{aligned} C_\alpha(u_1, u_2) \quad &= \quad (1 - [(1-u)^\alpha + (1-v)^\alpha \\ &\quad\quad \times - (1-u)^\alpha (1-v)^\alpha]^{1/\alpha}) \end{aligned}$$

An illustration of the association structure can be found in Figure 2.3.

### 2.2.5 Frank Model

Finally, the copula specification for the Frank model is given by:

$$C_\alpha(u_1, u_2) = -\frac{1}{\alpha} \log\left(1 + \frac{(e^{\alpha u_1} - 1)(e^{\alpha u_2} - 1)}{-1 + e^{-\alpha}}\right) \tag{2.8}$$

An illustration of the association structure can be found in Figure 2.4.

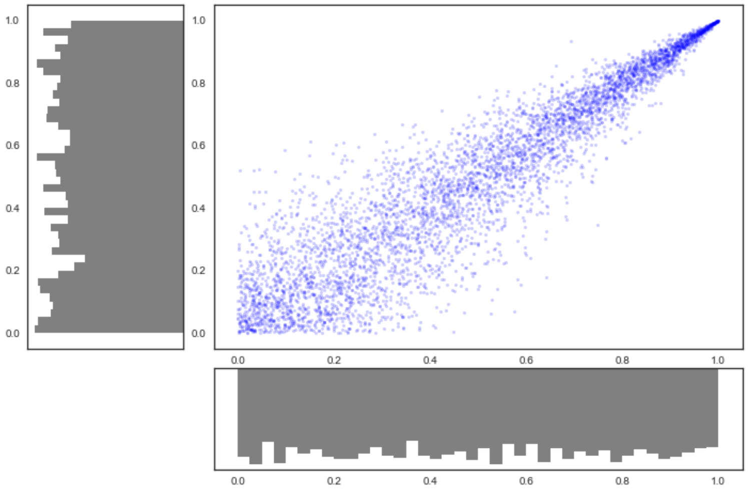


Figure 2.3: Copula Generation for Joe Model

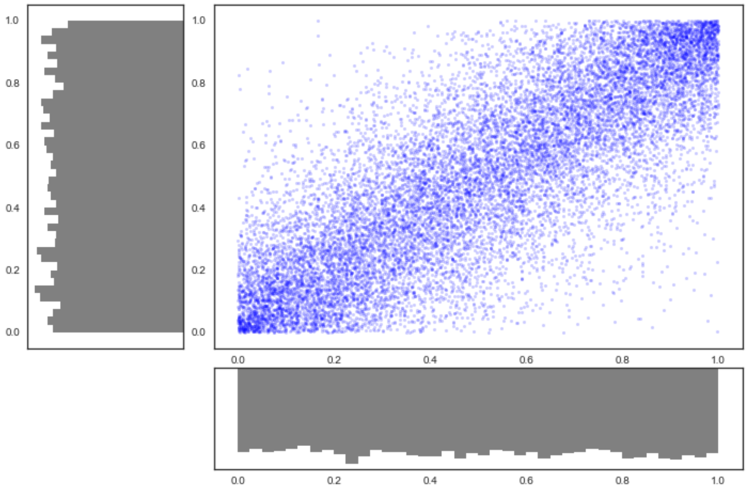


Figure 2.4: Copula Generation for Frank Model

## 2.3 Model and Likelihood Formulations

In this section, we develop the likelihood formulation for the bivariate survival models. Consider bivariate pairs of data $(T^*_{i1}, T^*_{i2})$ representing the true time to event for the $i^{th}$ individual in the study. Now, consider censoring times $(C_{i1}, C_{i2})$ and corresponding indicator functions $\delta_{ik} = I(C_{ik} \geq T^*_{ik})$, $k = 1, 2$. Then the observed times are given by $T_{ik} = \min(C_{ik}, T^*_{ik}), k = 1, 2$. The construction of these models can be conceptualized in a modular fashion; we begin by defining the marginal survival functions for each dimension and then subsequently define the joint distribution in terms of these marginal distributions. Our goal is to define a copula-based transformation joint model for the survival probability $P(T_1 > t_1, T_2 > t_2|X)$ using

$$S_k(t_k|X_i) = \left\{1 + r_k\Lambda_{0k}(t_k)\exp(\beta_k^{'}X_i)\right\}^{-r_k^{-1}} \tag{2.9}$$

with density

$$f_k(t_k|X) = \lambda_{0k}(t_k)\exp(\beta_k^{'}X_i)\left\{1 + r_k\Lambda_{0k}(t_k)\exp(\beta_k^{'}X_i)\right\}^{-(1+r_k^{-1})}. \tag{2.10}$$

For bivariate data, there are four combinations of possible observed indicators for the individual pairs of times. The likelihood component for two right censored observations is given by the joint survival function itself; which is given by the copula evaluated at the conditional marginal survival functions. This may be expressed as

$$P(T_1 > t_1, T_2 > t_2) = C_\alpha\left(\left\{1 + r_1\Lambda_{01}(t_1)\exp(\beta_1^{'}X_i)\right\}^{-r_1^{-1}}, \left\{1 + r_2\Lambda_{02}(t_2)\exp(\beta_2^{'}X_i)\right\}^{-r_2^{-1}}\right). \tag{2.11}$$

Next, when both times correspond to observed failures, the likelihood contribution is given by

$$P(T_1 = t_1, T_2 = t_2) \;=\; \frac{\partial^2 C_\alpha\left(\left\{1 + r_1\Lambda_{01}(t_1)\exp(\beta_1^{'}X_i)\right\}^{-r_1^{-1}}, \left\{1 + r_2\Lambda_{02}(t_2)\exp(\beta_2^{'}X_i)\right\}^{-r_2^{-1}}\right)}{\partial t_1 \partial t_2}$$

For the case that only one event occurs, we differentiate the copula function in the direction of the observed time. With out loss of generality consider the case that $T_1$

is censored and $T_2$ is observed. Then we have

$$P(T_1 > t_1, T_2 = t_2) \quad = \quad -\frac{\partial C_\alpha \left( \left\{1 + r_1 \Lambda_{01}(t_1) \exp(\beta_1' X_i)\right\}^{-r_1^{-1}}, \left\{1 + r_2 \Lambda_{02}(t_2) \exp(\beta_2' X_i)\right\}^{-r_2^{-1}} \right)}{\partial t_2}$$

Our likelihood for the bivariate data is then simply the product of the indicator combinations

$$\begin{aligned} L(\alpha, \beta_1, \beta_2, \lambda_1, \lambda_2, r_1, r_2 | X) &= \prod_{i=1}^{n} P(T_1 = t_1, T_2 = t_2)^{\delta_1 \delta_2} P(T_1 > t_1, T_2 = t_2)^{(1-\delta_1)\delta_2} \\ &\quad \times P(T_1 = t_1, T_2 > t_2)^{(1-\delta_2)\delta_1} P(T_1 > t_1, T_2 > t_2)^{(1-\delta_1)(1-\delta_2)} \\ &= \prod_{i=1}^{n} (c_\alpha(S_{i1}(t_1|X), S_{i2}(t_2|X)) f_{i1}(t_1|X) f_{i2}(t_2|X))^{\delta_1 \delta_2} \\ &\quad \times \left( -\frac{\partial C_\alpha\left(S_{i1}(t_1|X), S_{i2}(t_2|X)\right)}{\partial S_{i1}(t_1|X)} \cdot (-f_{i1}(t_1|X)) \right)^{\delta_{i1}(1-\delta_{i2})} \\ &\quad \times \left( -\frac{\partial C_\alpha(S_{i1}(t_1|X), S_{i2}(t_2|X))}{\partial S_{i2}(t_2|X)} \cdot (-f_{i2}(t_2|X)) \right)^{\delta_{i2}(1-\delta_{i1})} \\ &\quad \times C_\alpha(S_{i1}(t_1|X), S_{i2}(t_2|X))^{(1-\delta_1)(1-\delta_2)}. \end{aligned} \tag{2.12}$$

For statistical inference, the log-likelihood of this expression is used for computational feasibility.

Now that we have defined the likelihood of this model, computing the posterior distribution is done through Bayes' theorem. Bayesian analyses require the specification prior distributions on the model parameters i.e., we need to specify $p(\alpha, \beta_1, \beta_2, \lambda_1, \lambda_2, r_1, r_2)$. We will leave specific distributional assumptions to subsequent sections, but the posterior distribution of the model parameters will be of the form

$$\begin{aligned} P(\alpha, \beta_1, \beta_2, \lambda_1, \lambda_2, r_1, r_2 | X) &\propto L(\alpha, \beta_1, \beta_2, \lambda_1, \lambda_2, r_1, r_2 | X) \\ &\quad \times P(\alpha, \beta_1, \beta_2, \lambda_1, \lambda_2, r_1, r_2). \end{aligned} \tag{2.13}$$

In this paper, we first consider a parametric Weibull formation of the hazard, which exhibits a monotone hazard over time and is especially significant because it may be parameterized in a proportional hazards or accelerated failure time specification (Klein and Moeschberger, 2005). Mathematically, the baseline hazard takes the form $\lambda_k(t) = \lambda_k \rho_k t^{\rho_k}$ and the cumulative hazard function is given by $\Lambda_k(t) = \lambda_k t^{\rho_k}$.

## 2.4 Semi-parametric copula model

We now extend the preceding methodological development to a de-restricted form of a baseline hazard function $\lambda_0(t)$. By restricting oneself to a parametric hazard, we are limited to only consider monotonic hazards. Indeed, it is often the case that the true underlying hazard is non-monotone. Naturally, there are some parametric alternatives to a Weibull hazard, including the log-Normal distribution (Klein and Moeschberger, 2005), however, one alternative way to specify the baseline hazard is through the introduction of a non-linear spline function De Boor et al. (1978).

A spline function is nothing more than a linear combination of basis vectors (the exact basis will be discussed more in the subsequent section). Early implementations of a spline baseline hazard include Herndon and Harrell Jr (1995) in terms of the a truncated power basis. In this paper, we utilize the B-spline basis of De Boor et al. (1978).

### 2.4.1 Likelihood specification

In this section, we develop the inferential specification of the likelihood component of this model. Splines leverage the fact that a functional form may be expressed as a linear combination of a set of basis functions; specifically, the basis functions are generated conditionally on a set of (in this paper, fixed) knot points. Estimation of the underlying baseline hazard function therefore becomes an estimation procedure for the coefficients of the linear combination (for whatever basis is used). Although the *linear* component of a survival model may be replaced with splines to specify a non-linear covariate effect (by way of a linear construction), we focus here only on the development of the baseline hazard. Similarly to Sharef et al. (2010), we specify the spline function to model the baseline hazard function (although we do not adaptively select knot points as in the aforementioned paper). As an alternative, several authors advocate the modeling of the log-hazard, which is given by $\log \lambda_0(t) = \sum_{s=1}^{S} \alpha_s B_{s,q}(t)$. This formulation allows for the introduction of certain penalization methods on the linear coefficients, due to the fact that the log-hazard is not required to be positive. However, one important quantity of interest in our models is the cumulative hazard for the right censored observations. The log-hazard specification of the baseline hazard can be numerically integrated to give the cumulative hazard; see Davis and Rabinowitz (2007) implementation of quadrature integration. While numerical integration is feasible for smaller datasets, the computational complexity involved in fitting increasingly complex baseline hazard, coupled with the complexity of evaluating the likelihood motivates the development of a spline specification that leads to a closed-form solution for the baseline-cumulative hazard that is in terms of the coefficients used to parameterize the hazard function. The introduction of a closed form integral allows for easier gradient computations in the underlying computer software. Complexity is reduced by eliminating the need for numerical approximation and information is incorporated about the spline coefficients through the cumulative hazard.

The primary reason for this restriction is because the hazard itself is unrestricted in its shape, whereas the cumulative baseline hazard is strictly increasing over time; this fact motivates non-linearly modeling the hazard instead of the cumulative baseline hazard. Although there exist closed-form expressions of the derivative function of a specified spline, we will instead leverage the closed form expression for integration of B-splines instead. Following the model specification in Equation 2.9 , we see that it is necessary to compute the following quantity:

$$\Lambda_{0k}(t_k|\vec{\alpha}_k) = \int_0^{t_k} \lambda_{0k}(x|\alpha)dx. \tag{2.14}$$

Now, we wish to use the spline representation of baseline hazard which assumes that

$$\lambda_{0k}(t|\vec{\alpha}_k) = \sum_{s=1}^{S} \alpha_{sk} B_{s,q}(t_k) \tag{2.15}$$

where $q$ is the order of the spline $(degree + 1)$ over a set of knots $\boldsymbol{\xi}$, and $S = q + \#$ of knots. This is a simple dot product of $\boldsymbol{\alpha}$ and a vector containing the appropriate basis values at the observed time, $t_i$.

Now that we have specified the form of the baseline hazard, we now show the form of the baseline hazard. This quantity, is of course, the integrated (or cumulative) hazard function. As given in De Boor et al. (1978), the integral of a spline may be expressed as

$$\begin{aligned}
\Lambda_0(t|\vec{\alpha}) &= \int_0^t \lambda_0(x|\vec{\alpha})dx \\
&= \int_0^t \sum_{s=1}^{S} \alpha_s B_{s,q}(x)dx \\
&= \sum_{s=1}^{S-1} \left( \sum_{j=1}^{s} \alpha_j (\xi_{j+q} - \xi_j)/q \right) B_{s,q+1}(t). \qquad (2.16)
\end{aligned}$$

Hence, the full likelihood for a given subject the univariate model, is given by

$$\begin{aligned}
l_i(\beta, \alpha|X_i, t_i, \delta_i) &= \delta_i \left[ \log \left( \sum_{s=1}^{S} \alpha_s B_{s,q}(t_k) \right) + X_i^T \beta \right] \\
&- \left[ \sum_{s=1}^{S-1} \left( \sum_{j=1}^{s} \alpha_j (\xi_{j+q} - \xi_j)/q \right) B_{s,q+1}(t) \right] \exp(X_i^T \beta)
\end{aligned}$$

In figures 2.5 and 2.6, we can see one MCMC run on a simulated dataset to demonstrate this approach.

### 2.4.2 Computer Implementation

Although Equation 2.16 gives a closed form solution for the integral of the baseline hazard, this expression is not trivial in its computer implementation. Recall that we wish to leverage a tensor-based deep learning library to perform expensive gradient calculations for procedures such as Hamiltonian Monte Carlo, or alternative fitting procedures such as the Laplace's approximation (Shun and McCullagh, 1995) and variational Bayesian inference (Blei et al., 2017). A preliminary implementation of the integration task can be constructed using a nested loop (this is present in the 'IntegrateBs' R package). However, since we would like to pass arbitrary vectors as data into the model, we need to express the integral in terms of matrix algebra. Such steps will alleviate much computational overhead and will reduce the computational complexity of the differentiation for the fitting process.

Notably, we would like for theano, (or any deep learning library for that matter), to handle the spline coefficients as a tensor of arbitrary size and have all computations involve that specific computational tensor; notably $\vec{\alpha}$. Revisiting the summation in Equation 2.16, we have

$$\sum_{s=1}^{S-1} \left( \sum_{j=1}^{s} \alpha_j (\xi_{j+q} - \xi_j)/q \right) B_{s,q+1}(t) \tag{2.17}$$

for $\xi_0 \leq t \leq \xi_s$. One might notice that this expression constitutes a broadcasted dot product over an upper triangular matrix. To illustrate, we first consider the inner summation given by

$$\frac{1}{q} \sum_{j=1}^{k} \alpha_j (\xi_{j+q} - \xi_j). \tag{2.18}$$

For a given $k$, we notice that

$$
\begin{aligned}
\alpha_{integral,k} &= \frac{1}{q}\sum_{j=1}^{k}\alpha_j(\xi_{j+q}-\xi_j) && (2.19)\\
&= \frac{1}{q}\boldsymbol{\alpha}\cdot\begin{bmatrix}\xi_{1+q}-\xi_1\\ \xi_{2+q}-\xi_2\\ \vdots\\ \xi_{k+q}-\xi_k\\ \mathbf{0}_{(\#knots+q-k)\times 1}\end{bmatrix}_{(\#knots+q)\times 1} && (2.20)
\end{aligned}
$$

which is simply a scalar. Collecting these terms in a vector, say $\boldsymbol{\alpha_{integral}}$ which is of dimension $(s-1)\times 1$. This vector may be computed by broadcasting $\vec{\alpha}$ over the following matrix

$$
\begin{bmatrix}
\xi_{1+q}-\xi_1 & \xi_{1+q}-\xi_1 & \dots & \dots & \xi_{1+q}-\xi_1\\
0 & \xi_{2+q}-\xi_2 & \dots & \dots & \xi_{2+q}-\xi_2\\
\vdots & 0 & \vdots & \ddots & \xi_{3+q}-\xi_3\\
\vdots & \vdots & \vdots & \ddots & \vdots\\
0 & 0 & 0 & \dots & 0
\end{bmatrix}
$$

Further, we define the vector of evaluated $S-1$ basis functions $\boldsymbol{B}_{q+1}(t)$. We see then that

$$
\sum_{s=1}^{S-1}\left(\sum_{j=1}^{s}\alpha_j(\xi_{j+q}-\xi_j)/q\right)B_{s,q+1}(t)=\boldsymbol{\alpha_{integral}}\cdot\boldsymbol{B}_{q+1}(t) \tag{2.21}
$$

A comparison of the accuracy and speed of the implementation can be found in the Appendix.

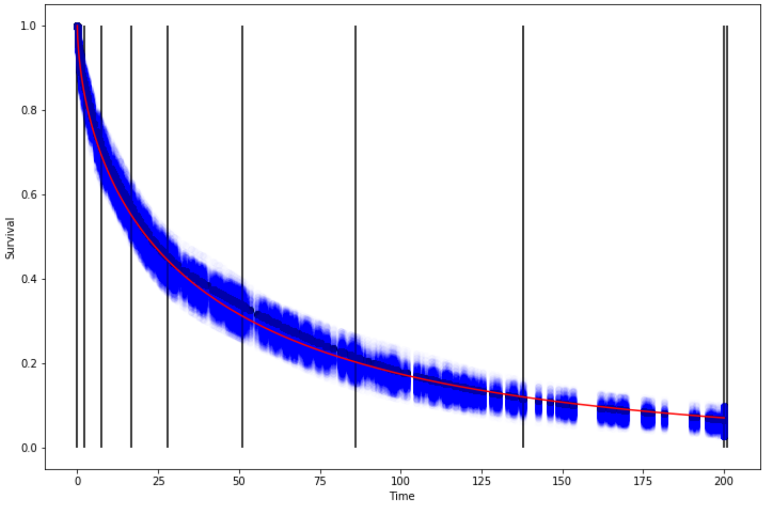


Figure 2.5: Cumulative hazard function using univariate B-splines on the baseline hazard. Blue points represent the posterior MCMC draws, the black vertical lines represent the knot locations, and the red line represents the true simulated cumulative hazard function.

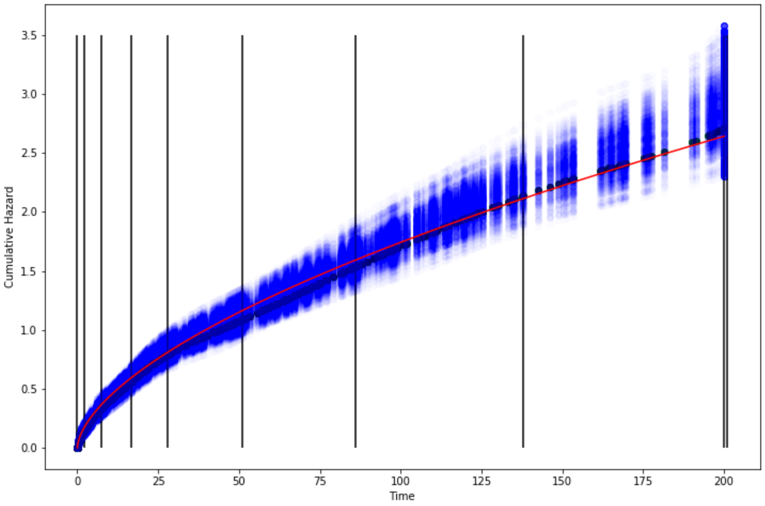


Figure 2.6: Survival function using univariate B-splines on baseline hazard. Blue points represent the posterior MCMC draws, the black vertical lines represent the knot locations, and the red line represents the true simulated survival function.

### 2.4.3 Multivariate Censored Likelihood

The development of the likelihood function for the multivariate spline copula model proceeds in a similar fashion to the parametric model. The bivariate copula under this specification has the same form as Equation 4.1 with $\Lambda_0(t)$ and $\lambda_0(t)$ replaced by 2.16 and 2.15, respectively.

## 2.5 Inference

In Bayesian analysis, the most common strategy for computing posterior distributions and quantities of interest is through Markov Chain Monte Carlo (MCMC) wherein an ergodic Markov Chain is constructed in such a manner that its stationary distribution is the posterior distribution of interest, $P(\theta|X)$ (Brooks et al., 2011). With regards to Bayesian sampling mechanisms for survival copulas, Romeo et al. (2006) implemented a Gibbs sampler for proportional hazards copula models using the BUGS software Recent advances in the MCMC literature have led to the development of more efficient sampling schemes including Hamiltonian Monte Carlo and No U-Turn Sampling (Gelman et al., 2014). Certainly, the area of Bayesian survival analysis has a multitude of computational development of Gibbs sampling methods developed for specific models (Ibrahim et al., 2005). In the first part of these analyses, we will use No-U-turn sampling with pymc3 (Salvatier J, 2016). Pymc3 is a general purpose probabilistic programming package written in the Python computing language.

## 2.6 Simulations

To assess the statistical properties of our proposed methodology, we implement a series of simulations to determine the overall efficacy of the approach. For the simulations, we generated data from non-proportional hazards models of the form seen in Equation 2.9. Simulation strategies for proportional hazards may be found in

Bender et al. (2005) and we extend this methodology to the general transformation model previously outlined. Simulation from copula models is a somewhat challenging undertaking, depending on the type of copula that is being simulated. The generalized approach, however, is described by Marshall and Olkin (1988). For these simulation studies, however, the R package 'copula' was used to simulate from the underlying copula. After simulating from the copula models, we then took each margin (which is uniformly distributed), and simulated draws from the joint model using the inverse CDF of the transformation model. We simulated cohort covariates for that acted as proxies for age and sex. Specifically, for the age covariate, we drew $X_{age} \sim Gamma(\alpha = 10, \beta = .3)$ and sex was generated by a simple binomial variable with success probability $p = .5$. For our simulations we set the transformation parameters to $r_1, r_2 = 0.5$. For the parametric models, the baseline hazard and frailty parameters were given Half-Cauchy(2.5) distributions, while diffuse Normal distributions were used for the regression coefficients (zero mean and variance of $100^2$). Note that our modeling framework allows for any specification of prior distribution and we are not restricted to a specific form.

.

The basic graphical workflow of our approach is illustrated in Figure 2.7. We can see that each margin of the copula model is compartmentalized and is joined together via a copula function $C_\alpha$.

### 2.6.1 Initial Values

In principle, the fitting of a Bayesian model proceeds by implementing Markov Chain. Theoretically, as long as the sampling process has sufficient time to sample, draws from the stationary distribution of the Markov Chain will be simulations from the target posterior distribution. However, in practice, MCMC can be highly sensitive to initial values. Indeed, in our initial attempts for simulations, we found that using

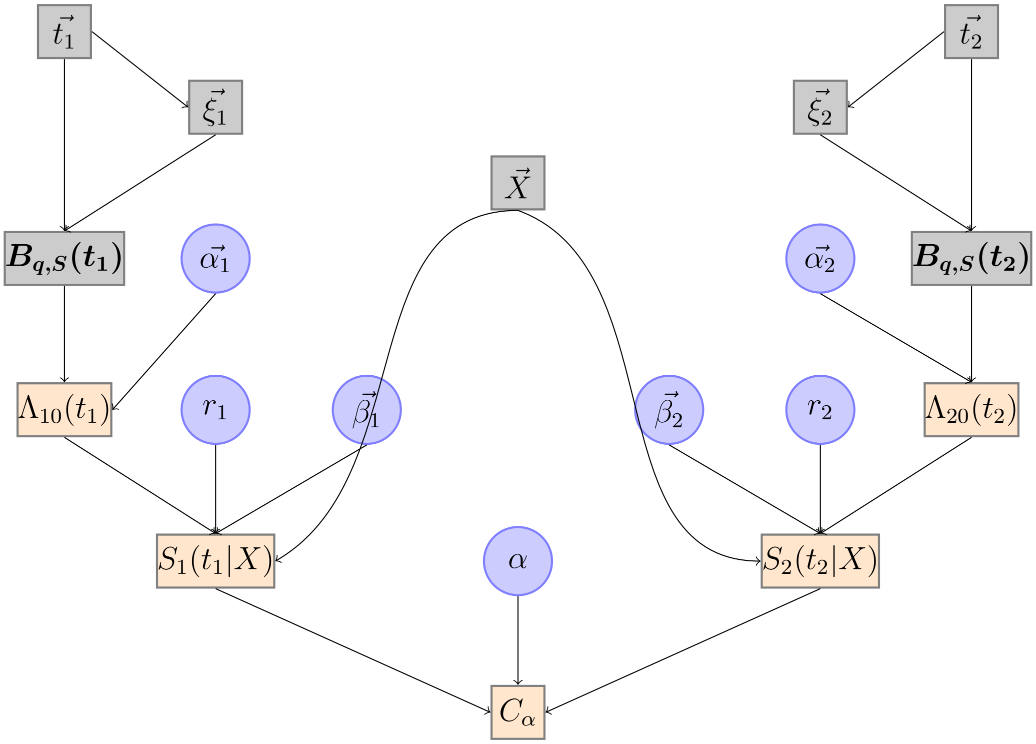


Figure 2.7: Graphical construction of parametric copula model. The grey squares represent quantities in the computational graph corresponding to data/data derived quantities, purple circles represent random variables, and orange rectangles represent deterministic (or derived) variables in the computational graph.

default values of mean on the prior distributions resulted in failure of the initial energy of the Hamiltonian Monte Carlo. This motivated the use of better initial values.

For the Frank, Joe, and Gumbel models, an initial implementation of initial values was via the usage of the maximum likelihood estimates of the marginal distributions. Notably, for these models, we performed an initial optimization of the marginal survival parameters with Nelder-Mead optimization. Subsequently, we optimized the joint distribution to obtain an initial point for association parameter. Using the optimization in this last step, we initialized the MCMC sample. For the simulated models, this initialization was mostly sufficient, however, due to the unconstrained nature of the Nelder-Mead algorithm, it was possible for the optimizer to return negative values for the transformation parameter $r$. This ultimately led the sampler to occasionally fail due to the initial value lying outside of the support of the distribution. However, we found that if the MCMC sampler was able to sample at least one sample, then the simulation would run the entire time. For numerical stability, we recommend either constraining the optimization or just using the mean of the posterior distribution from MCMC marginal fits as starting values for the full joint distribution fit. For the Clayton models contained throughout, the full posterior distribution was maximized (*maximum a posteriori*) and was used as an initial value. However, this joint optimization occasionally led to some failures in initializing the MCMC sampling. We will discuss this fact and an alternately proposed initialization scheme in the next section.

### 2.6.2 Simulations

In this section, we present the results of simulation studies performed on simulated datasets of size 5000. For the parametric Clayton model, we used 2000 Hamiltonian tuning steps and 5000 draws and we initialized the MCMC at the *maximum a-posteriori* estimate. For each other parametric copula, we used No-U-Turn Hamiltonian MCMC with 500 tuning steps and 5000 post-tuning steps as well as a two-staged maximum likelihood initialization (described in the previous section). Uniform censoring was introduced and censoring rates for the Clayton model were 10% and 14% for $T_1$ and $T_1$, respectively (note, although this censoring rate is low here, we perform simulations with much higher rates in the semi-parametric portion of the paper). Censoring rates for the Gumbel model were 18% and 25.1%, 17.5% and 24.4% for the Frank model, and 17.3% and 23.5% for the Joe model. For each parameter, we provide the average bias of the posterior mean (ABPM) over all simulations, the standard deviation of the posterior mean (SDPM), the mean standard deviations of the posterior distributions (SDS), the mean squared error, and the frequentist coverage probability (CP) of the 95% quantile credible interval. For the reader's convenience, we multiplied each of these quantities by 100.

Interpreting these results, conditional on an effective starting value, we see that with diffuse priors on the parameters, the coverage probabilities of the simulations are between 93.0% to 97.2% with most parameters reaching the nominal level of 95%. Overall, we see low average bias and low MSE in the simulations for the $\beta$ effect sizes; the highest of which in absolute value is $5.03 \times 10^{-3}$ corresponding to the $\beta_{21}$ estimate in the Frank model. Overall, the highest observed bias for the posterior mean was the association parameter $\alpha$ across all of the models (the highest of which was $1.2 \times 10^{-2}$). We notice that the standard deviation of the posterior samples for the transformation parameter and for the copula association parameter are the largest of all parameters indicating that there is less sample information about these parameters than for the effect sizes. Additionally, the diffusion in these posterior samples likely contributes to the increase in bias of the posterior mean. It should be noted to the reader that these simulation results are based on runs that had reasonable initial values that allowed the MCMC sampler to run. In some cases, the maximization attempt for the initial value (that is the maximization of the full joint posterior as

| Copula | Param. | Truth | ABPM | SDPM | SDS | MSE | CP |
|---|---|---|---|---|---|---|---|
| | | | (×100) | (×100) | (×100) | (×100) | (×100) |
| Clayton | $\beta_{11}$ | 1.63 | -0.144 | 4.775 | 4.604 | 0.228 | 94.2 |
| | $\beta_{12}$ | 0.03 | -0.009 | 0.173 | 0.167 | $< 0.001$ | 93.0 |
| | $\beta_{21}$ | 0.8 | -0.246 | 3.881 | 3.717 | 0.151 | 95.2 |
| | $\beta_{22}$ | 0.03 | -0.017 | 0.172 | 0.168 | $< 0.001$ | 94.6 |
| | $\alpha$ | 3.21 | 1.288 | 7.992 | 8.162 | 0.655 | 94.6 |
| | $\lambda_1$ | 0.0047 | 0.006 | 0.052 | 0.051 | $< 0.001$ | 93.8 |
| | $\lambda_2$ | 0.0037 | 0.106 | 0.052 | 0.051 | $< 0.001$ | 94.0 |
| | $\rho_1$ | 0.716 | 0.036 | 1.330 | 1.376 | 0.018 | 95.8 |
| | $\rho_2$ | 0.725 | -0.022 | 1.508 | 1.439 | 0.023 | 94.2 |
| | $r_1$ | 0.5 | 0.219 | 3.504 | 3.657 | 0.123 | 95.0 |
| | $r_2$ | 0.5 | 0.206 | 4.264 | 3.968 | 0.182 | 94.8 |
| Joe | $\beta_{11}$ | 1.3 | -0.085 | 4.294 | 4.032 | 0.184 | 93.2 |
| | $\beta_{12}$ | 0.03 | -0.009 | 0.159 | 0.165 | $< 0.001$ | 96.4 |
| | $\beta_{21}$ | 0.8 | -0.034 | 3.827 | 3.627 | 0.147 | 93.4 |
| | $\beta_{22}$ | 0.03 | -0.007 | 0.163 | 0.165 | $< 0.001$ | 95.6 |
| | $\alpha$ | 9 | 0.618 | 20.561 | 19.855 | 4.231 | 94.4 |
| | $\lambda_1$ | 0.0047 | 0.006 | 0.048 | 0.049 | $< 0.001$ | 95.6 |
| | $\lambda_2$ | 0.0037 | 0.106 | 0.048 | 0.049 | $< 0.001$ | 96.0 |
| | $\rho_1$ | 0.716 | 0.037 | 1.297 | 1.319 | 0.017 | 95.8 |
| | $\rho_2$ | 0.725 | 0.059 | 1.337 | 1.353 | 0.018 | 97.2 |
| | $r_1$ | 0.5 | 0.249 | 4.465 | 4.474 | 0.200 | 94.6 |
| | $r_2$ | 0.5 | 0.450 | 5.076 | 4.977 | 0.260 | 95.6 |
| Gumbel | $\beta_{11}$ | 1.3 | -0.311 | 4.684 | 4.675 | 0.22 | 94.2 |
| | $\beta_{12}$ | 0.03 | -0.033 | 0.201 | 0.197 | $< 0.001$ | 94.2 |
| | $\beta_{21}$ | 0.8 | -0.147 | 4.397 | 4.299 | 0.194 | 95.6 |
| | $\beta_{22}$ | 0.03 | -0.028 | 0.199 | 0.198 | $< 0.001$ | 94.2 |
| | $\alpha$ | 7.9 | 1.700 | 14.216 | 14.639 | 2.050 | 96.8 |
| | $\lambda_1$ | 0.0047 | 0.015 | 0.052 | 0.053 | $< 0.001$ | 95.4 |
| | $\lambda_2$ | 0.0037 | 0.115 | 0.052 | 0.053 | $< 0.001$ | 95.4 |
| | $\rho_1$ | 0.716 | -0.135 | 1.282 | 1.338 | 0.017 | 96.0 |
| | $\rho_2$ | 0.725 | -0.093 | 1.350 | 1.373 | 0.018 | 94.6 |
| | $r_1$ | 0.5 | -0.175 | 4.340 | 4.314 | 0.189 | 95.0 |
| | $r_2$ | 0.5 | 0.227 | 4.785 | 4.747 | 0.230 | 95.8 |
| Frank | $\beta_{11}$ | 1.3 | -0.283 | 4.821 | 4.717 | 0.233 | 93.6 |
| | $\beta_{12}$ | 0.03 | -0.019 | 0.184 | 0.187 | $< 0.001$ | 94.8 |
| | $\beta_{21}$ | 0.8 | -0.503 | 4.113 | 4.263 | 0.172 | 96.0 |
| | $\beta_{22}$ | 0.03 | -0.027 | 0.183 | 0.192 | $< 0.001$ | 95.2 |
| | $\alpha$ | 8 | 0.848 | 14.416 | 14.849 | 2.085 | 95.6 |
| | $\lambda_1$ | 0.0047 | 0.009 | 0.056 | 0.055 | $< 0.001$ | 94.8 |
| | $\lambda_2$ | 0.0037 | 0.109 | 0.056 | 0.055 | $< 0.001$ | 96.6 |
| | $\rho_1$ | 0.716 | 0.071 | 1.627 | 1.561 | 0.027 | 93.6 |
| | $\rho_2$ | 0.725 | -0.045 | 1.748 | 1.724 | 0.031 | 94.6 |
| | $r_1$ | 0.5 | 0.292 | 5.608 | 5.382 | 0.315 | 93.0 |
| | $r_2$ | 0.5 | 0.242 | 6.928 | 6.835 | 0.481 | 95.0 |

Table 2.1: Copula simulation results for 500 simulations for parametric Archimedean copulas, results are multiplied by 100

an initial value) failed to yield an estimate that allowed Hamiltonian Monte Carlo to tune and sample. We implemented an alternative initialization procedure that yielded less HMC initialization errors overall and yielded nearly identical results to those presented in table 2.1. Results for this followup simulation can be found in table 2.11 in the Appendix.

### 2.6.3 Semi-parametric Simulations

For the Clayton semi-parametric spline model, we performed a series of simulations for a variety of different spline specifications. Here, we present the results of a non-restricted spline with a simple diffuse Half-Cauchy prior, a more informative Half-Cauchy, and a hierarchical Half-Cauchy distribution on the spline coefficients. In these simulations we fixed 15 and 10 knots at evenly spaced sample quantiles of the observed data points.

### 2.6.4 Unpenalized Spline Model

As a preliminary benchmark, we fit an non-hierarchical, unpenalized spline model that used a Half Cauchy prior. Model 1 represents the model with a HalfCauchy(0.001) prior distribution and Model 2 represents the model with a HalfCauchy($\hat{b}$) prior distribution where $\hat{b}$ is the MAP point estimate. We the results of these simulations can be found in table 2.2.

We notice a slight under-coverage of the posterior credible intervals. This is likely due to the fact that the coefficients of the spline corresponding to the control points lie outside of the space of the observed data.

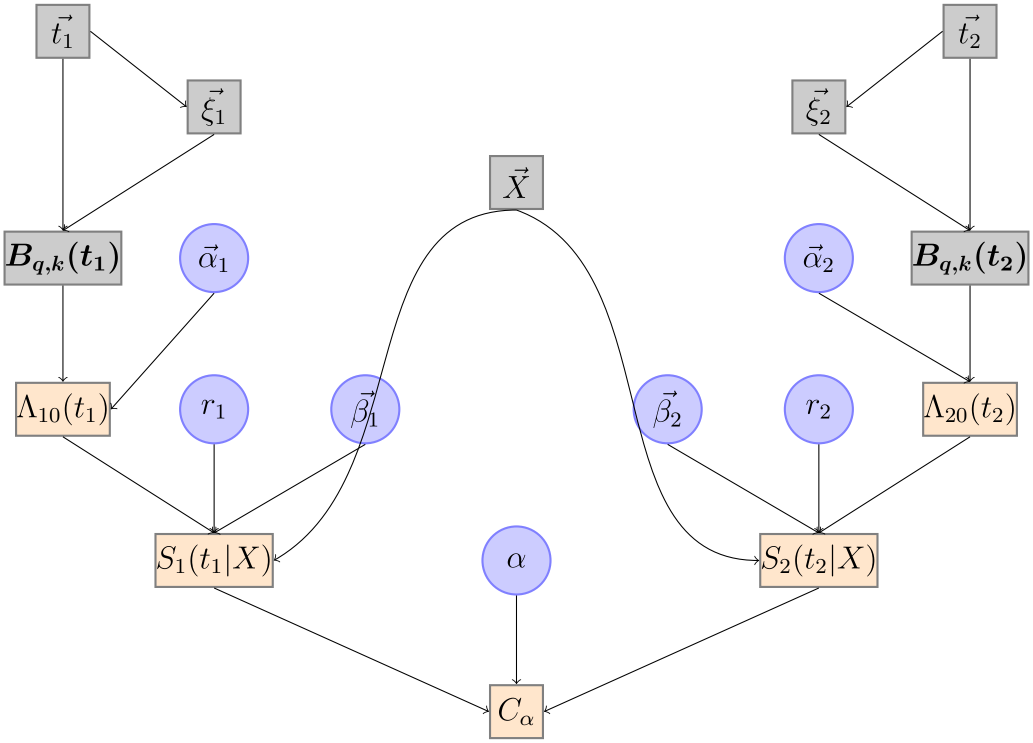


Figure 2.8: Graphical construction of spline-based copula model. The grey squares represent quantities in the computational graph corresponding to data/data derived quantities, purple circles represent random variables, and orange rectangles represent deterministic (or derived) variables in the computational graph.

| Model | Param. | Truth | ABPM | SDPM | SDS | MSE | CP |
|---|---|---|---|---|---|---|---|
| | | | ($\times$100) | ($\times$100) | ($\times$100) | ($\times$100) | ($\times$100) |
| Model 1 | $\beta_{11}$ | 1.3 | 1.547 | 3.251 | 3.202 | 0.13 | 92.1 |
| | $\beta_{12}$ | 0.03 | -0.007 | 0.107 | 0.106 | $< 0.001$ | 94.3 |
| | $\beta_{21}$ | 0.8 | 1.225 | 2.955 | 2.821 | 0.102 | 92.1 |
| | $\beta_{22}$ | 0.03 | 0.012 | 0.115 | 0.117 | $< 0.001$ | 95.6 |
| | $\alpha$ | 3.21 | -0.989 | 4.453 | 4.699 | 0.208 | 94.7 |
| | $r_1$ | 0.5 | 2.592 | 3.294 | 3.367 | 0.176 | 91.2 |
| | $r_2$ | 0.5 | 3.575 | 4.662 | 4.494 | 0.345 | 86.8 |
| Model 2 | $\beta_{11}$ | -0.63 | -0.815 | 1.914 | 2.08 | 0.043 | 95.6 |
| | $\beta_{12}$ | 0.03 | -0.043 | 0.093 | 0.095 | $< 0.001$ | 93.6 |
| | $\beta_{21}$ | -0.69 | -0.213 | 1.845 | 1.983 | 0.034 | 96.0 |
| | $\beta_{22}$ | 0.02 | -0.054 | 0.073 | 0.074 | $< 0.001$ | 89.0 |
| | $\alpha$ | 3.21 | -0.044 | 3.424 | 3.582 | 0.117 | 95.8 |
| | $r_1$ | 0.5 | 1.998 | 6.067 | 6.219 | 0.408 | 94.1 |
| | $r_2$ | 0.5 | -0.349 | 4.2 | 4.174 | 0.178 | 95.8 |

Table 2.2: Initial simulations with unrestricted spline specification on the baseline hazard

### 2.6.5 Penalized Spline

We next consider a penalized version of the spline baseline hazard. If we consider the model implemented in the previous section, we see that it was necessary to specify the hyper-parameter of the prior distribution for the spline coefficient *a priori*. We saw that the coefficients corresponding to the control points that were placed just past the study end date tended to be pulled toward the prior mean due to the fact that no-data existed to inform the posterior distribution. To amend this, we consider penalizing the spline coefficients by introducing a hierarchical prior distribution on the hyper-parameter of the spline coefficient prior distribution. By introducing this prior distribution, the spline coefficients themselves are treated as random variables from some generative process endowed with a parameter learned from the data. Now, since we assume that the spline coefficients corresponding to the control points follow the same distribution as the spline coefficients corresponding to the interior knots, we can better infer the posterior due to borrowing of strength. In other words, we borrow strength from the inference on the interior knots to better infer the posterior of the control points due to the fact that the interior knots have data associated with them. The results of 500 simulations under this specification can be seen in tables 2.3, 2.4, 2.5, and 2.6. Figures 2.9, 2.10, 2.11, and 2.12, present marginal survival functions over several simulations (mean is in blue, and 95% quantiles are in red) using 15 knots. Figures 2.13, 2.14, 2.15, and 2.16 present simulations using 10 knots. In each of these figures, the black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

We can see that of the splines implementations, the penalized spline did better at recovering the true underlying model parameters than the unpenalized model; notably, the frequentist coverage probabilities were all were near the nominal level whereas the unpenalized implementations contained under-coverage in some param-

| Parameter | Truth | ABPM<br>(×100) | SDPM<br>(×100) | SDS<br>(×100) | MSE<br>(×100) | CP<br>(×100) |
|---|---|---|---|---|---|---|
| $\beta_{11}$ | -0.63 | -0.463 | 5.049 | 5.088 | 0.257 | 96.2 |
| $\beta_{12}$ | 0.03 | -0.014 | 0.214 | 0.235 | $< 0.001$ | 96.6 |
| $\beta_{21}$ | -0.69 | 0.466 | 4.663 | 4.985 | 0.220 | 95.8 |
| $\beta_{22}$ | 0.02 | -0.028 | 0.179 | 0.191 | $< 0.001$ | 95.2 |
| $\alpha$ | 3.21 | 0.554 | 9.793 | 9.487 | 0.962 | 95.2 |
| $r_1$ | 0.5 | 2.511 | 11.926 | 14.088 | 1.485 | 98.2 |
| $r_2$ | 0.5 | -0.370 | 9.109 | 10.243 | 0.831 | 96.8 |

Table 2.3: Penalized spline results with 10 knots and sample size of 5000, more censoring

| Parameter | Truth | ABPM<br>(×100) | SDPM<br>(×100) | SDS<br>(×100) | MSE<br>(×100) | CP<br>(×100) |
|---|---|---|---|---|---|---|
| $\beta_{11}$ | -0.63 | 0.565 | 4.268 | 4.518 | 0.185 | 95.4 |
| $\beta_{12}$ | 0.03 | -0.051 | 0.203 | 0.210 | $< 0.001$ | 94.6 |
| $\beta_{21}$ | -0.69 | 1.010 | 4.595 | 4.685 | 0.221 | 96.0 |
| $\beta_{22}$ | 0.02 | -0.055 | 0.181 | 0.180 | $< 0.001$ | 92.6 |
| $\alpha$ | 3.21 | 0.731 | 8.508 | 8.341 | 0.729 | 94.4 |
| $r_1$ | 0.5 | -1.173 | 7.971 | 8.832 | 0.649 | 96.8 |
| $r_2$ | 0.5 | -2.307 | 7.853 | 8.147 | 0.670 | 94.4 |

Table 2.4: Penalized spline results with 10 knots and sample size of 5000, less censoring

| Parameter | Truth | ABPM<br>(×100) | SDPM<br>(×100) | SDS<br>(×100) | MSE<br>(×100) | CP<br>(×100) |
|---|---|---|---|---|---|---|
| $\beta_{11}$ | -0.63 | -0.482 | 4.633 | 5.044 | 0.217 | 95.8 |
| $\beta_{12}$ | 0.03 | -0.011 | 0.231 | 0.234 | 0.001 | 94.4 |
| $\beta_{21}$ | -0.69 | 0.639 | 4.603 | 4.936 | 0.216 | 95.6 |
| $\beta_{22}$ | 0.02 | -0.046 | 0.181 | 0.188 | $< 0.001$ | 95.4 |
| $\alpha$ | 3.21 | 0.611 | 9.685 | 9.472 | 0.942 | 94.6 |
| $r_1$ | 0.5 | 1.916 | 11.762 | 13.862 | 1.420 | 98.4 |
| $r_2$ | 0.5 | -2.025 | 8.876 | 9.923 | 0.829 | 96.2 |

Table 2.5: Penalized spline results with 15 knots and sample size of 5000, more censoring

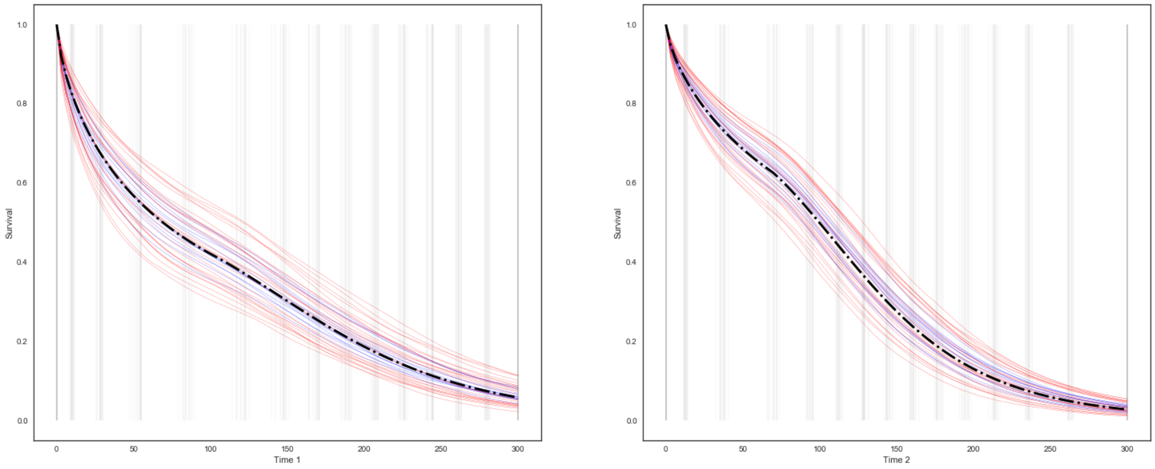


Figure 2.9: Plot of posterior marginal survival functions, male, 99 years old, 15 knots. The black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

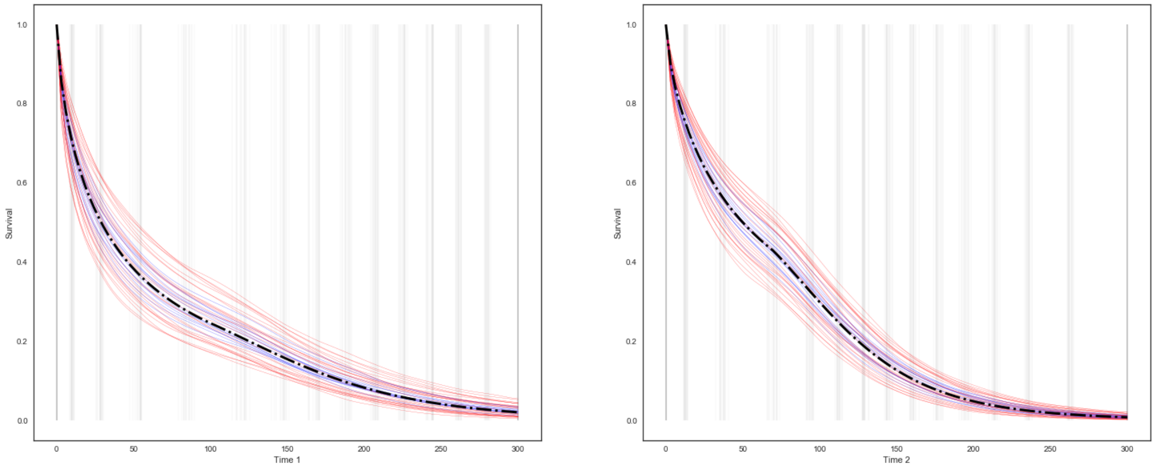


Figure 2.10: Plot of posterior marginal survival functions, female, 99 years old, 15 knots. The black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

eters. As a result, we conclude that the penalized spline model performs best with respect to the provided statistical properties.

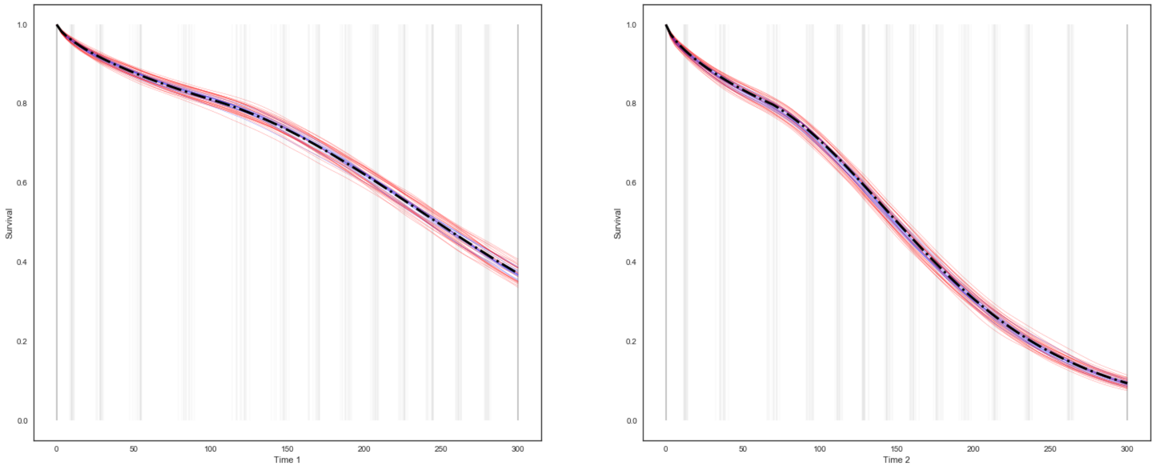


Figure 2.11: Plot of posterior marginal survival functions, female, 25 years old, 15 knots. The black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

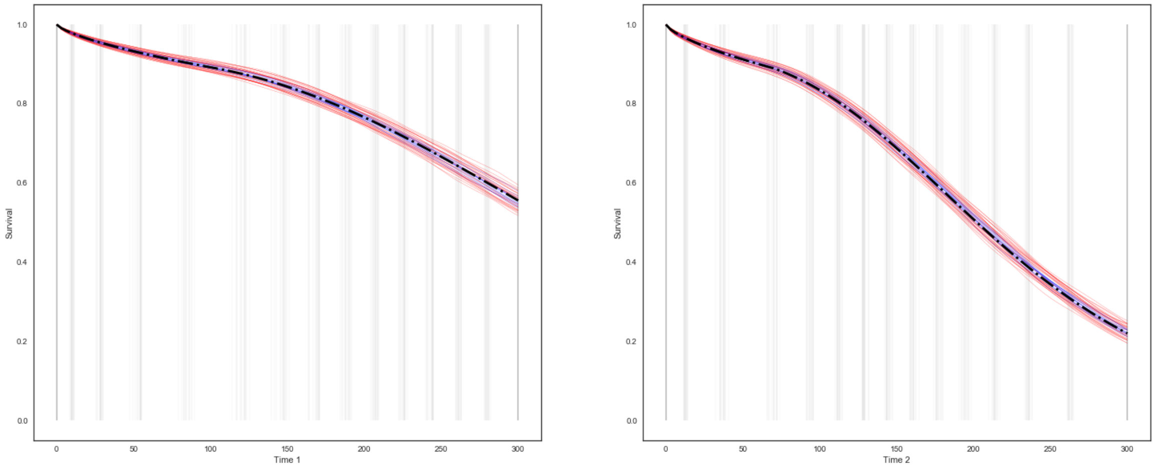


Figure 2.12: Plot of posterior marginal survival functions, male, 25 years old, 15 knots. The black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

Finally, figures 2.18 and present posterior distributions of joint probabilities (i.e. posterior copula values for the penalized spline model).

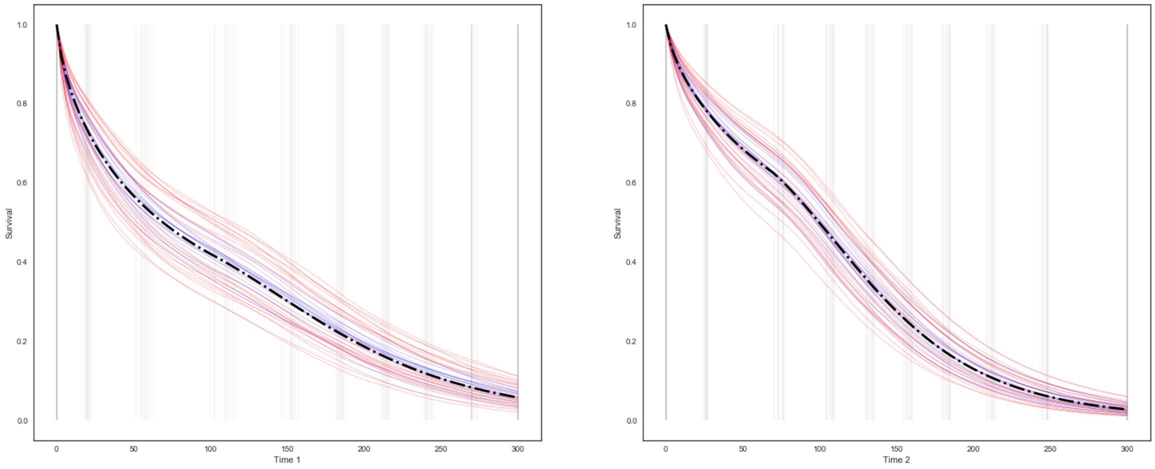


Figure 2.13: Plot of posterior marginal survival functions, male, 99 years old, 10 knots. The black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

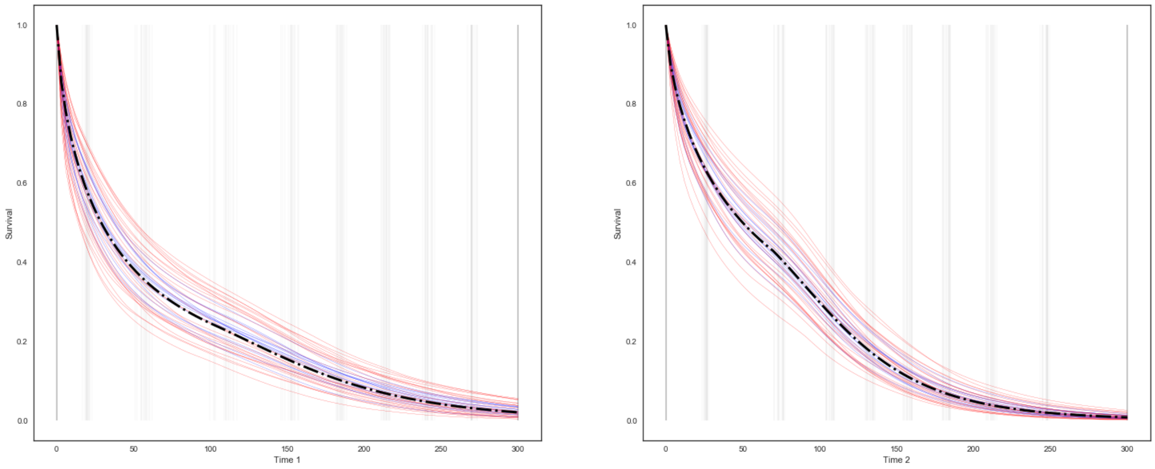


Figure 2.14: Plot of posterior marginal survival functions, female, 99 years old, 10 knots. The black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

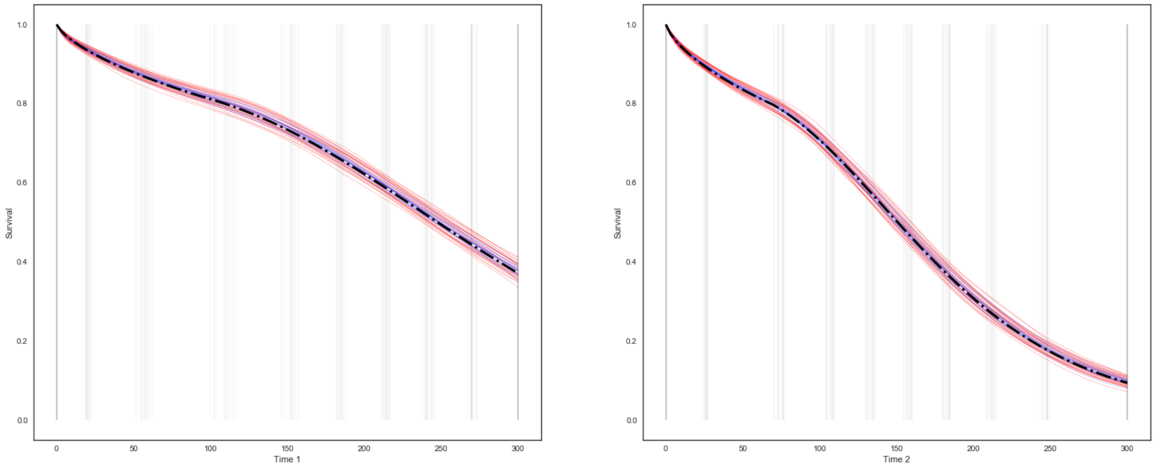


Figure 2.15: Plot of posterior marginal survival functions, female, 25 years old, 10 knots. The black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

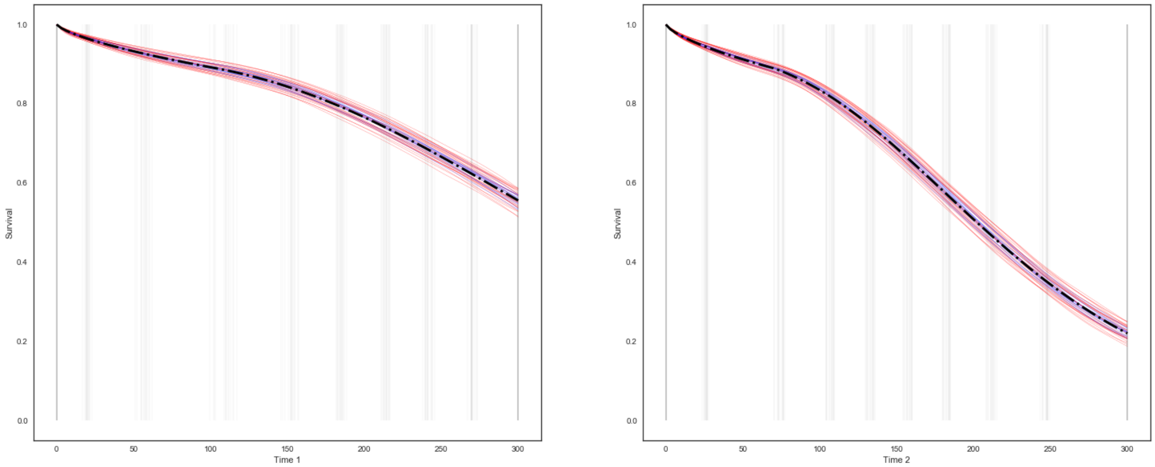


Figure 2.16: Plot of posterior marginal survival functions, male, 25 years old, 10 knots. The black dashed line represents the true survival functions, the red lines represent the 95% quantile bands, black vertical lines are knot locations, and the blue lines represent the means over each of the simulations.

## 2.7 Application

As an application of the methods described in this paper, we use the proposed methods to analyze a cohort from the Atherosclerosis Risk in Communities Study

| Parameter | Truth | ABPM | SDPM | SDS | MSE | CP |
|---|---|---|---|---|---|---|
| | | (×100) | (×100) | (×100) | (×100) | (×100) |
| $\beta_{11}$ | -0.63 | 1.282 | 4.132 | 4.44 | 0.187 | 94.8 |
| $\beta_{12}$ | 0.03 | -0.093 | 0.190 | 0.206 | $< 0.001$ | 93.6 |
| $\beta_{21}$ | -0.69 | 1.702 | 4.463 | 4.633 | 0.228 | 93.0 |
| $\beta_{22}$ | 0.02 | -0.078 | 0.172 | 0.178 | $< 0.001$ | 92.8 |
| $\alpha$ | 3.21 | 0.422 | 7.971 | 8.319 | 0.637 | 96.4 |
| $r_1$ | 0.5 | -3.458 | 7.721 | 8.582 | 0.716 | 94.4 |
| $r_2$ | 0.5 | -3.777 | 7.083 | 7.954 | 0.644 | 92.8 |

Table 2.6: Penalized spline results with 15 knots and sample size of 5000, less censoring

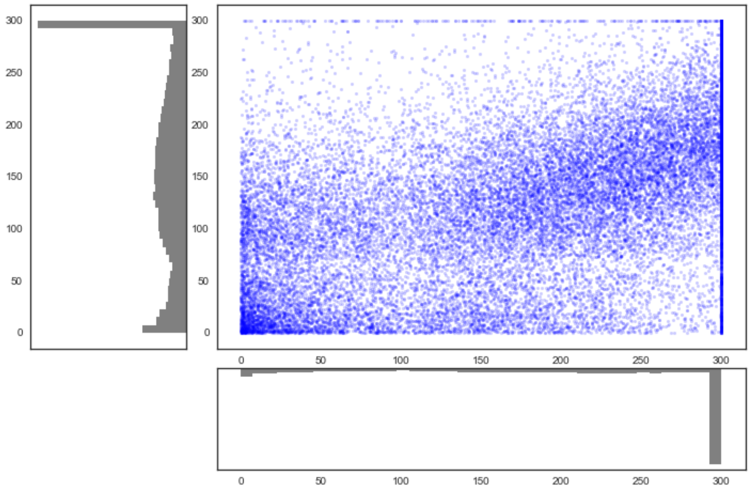


Figure 2.17: Simulated Bivariate Survival Times for Semi-Parametric Simulations

(Schmidt et al., 1999). We defined $T_1$ as the time to myocardial infarction, and $T_2$ as the time to stroke. Covariates that were included in the model included race, baseline age, sex, systolic blood pressure, diabetes, hypertension and smoking status. In the Clayton semi-parametric version of the model, sampling failed to initialize with the continuous covariates on their original scale. As such, we therefore standardized these values in the semi-parametric model.

We use this dataset to perform posterior estimation of joint survival given the covariates. The data we used contain 4639 patients who are more than 65 years of

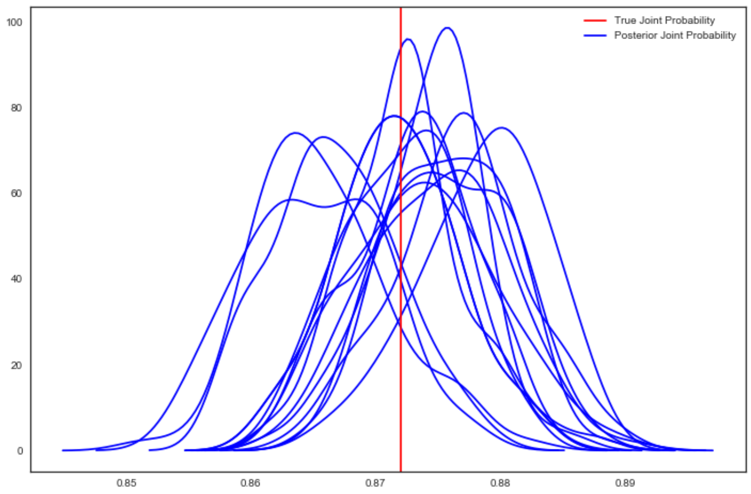


Figure 2.18: Joint survival probability for male 25 years old, 15 knots. Each blue kernel density plots represent the posterior distribution of the probability for a single simulation.

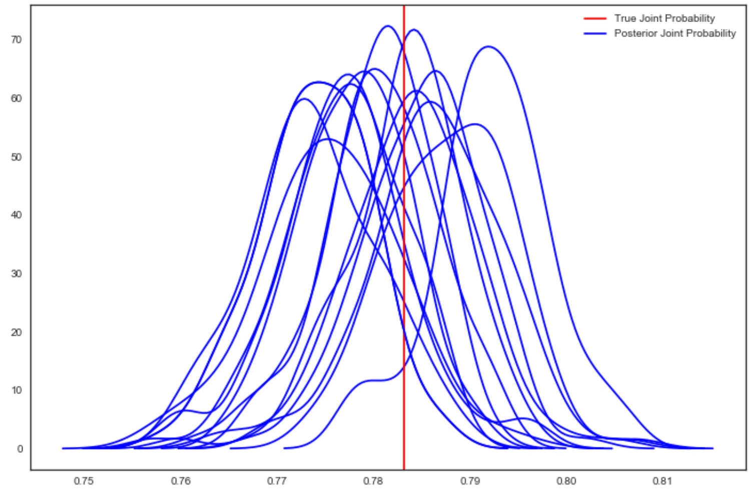


Figure 2.19: Joint survival probability for female 25 years old, 15 knots. Each blue kernel density plots represent the posterior distribution of the probability for a single simulation.

age. The censoring rates for myocardial infarction and time to stroke were 89.6% and 88.7%, respectively.

Figure 2.20 shows the bivariate relationship between the stroke and myocardial infarction times. Initial inspection of the scatterplot reveals a very strong linear relationship between the observed stroke times and the observed myocardial infarction times; indeed, 83.1% of the cohort had identical stroke and myocardial infarction times. However, among these 3854 individuals whose stroke and myocardial infarction times were identical, only 13 patients experienced both events; the rest of these observations were censored on at least one of the outcomes.

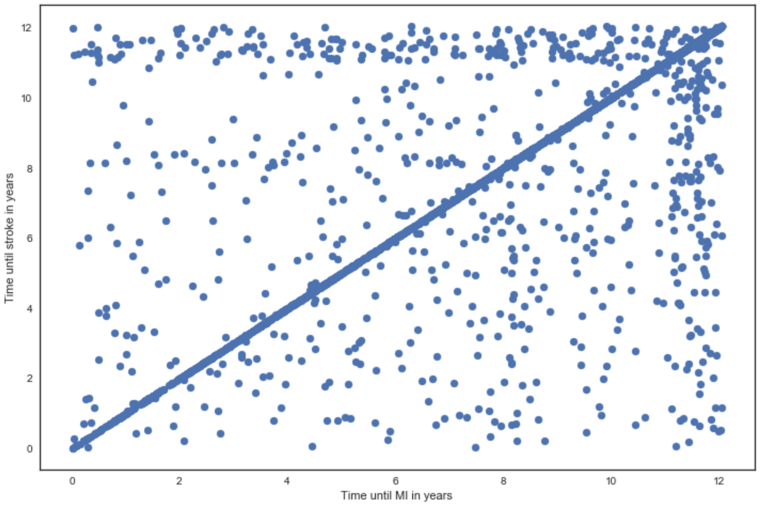


Figure 2.20: Bivariate observed times for ARIC dataset

For this application, we implemented the four previously discussed copula models on the dataset. Tables 2.8 and 2.9 present the model fits for each copula.

In table 2.7, we present the results of the WAIC (Vehtari et al., 2017) to compare the different copula models. The WAIC can be thought of as a large sample approximation to cross-validation, and is defined as

$$WAIC = -2\left(\sum_{i=1}^{n} \log\left(\frac{1}{S}\sum_{s=1}^{S} p(y_i|\theta^s)\right) - \sum_{i=1}^{n} V_{s=1}^{S}(\log(p(y_i|\theta^s)))\right) \tag{2.22}$$

where

$$V_{s=1}^{S} a_s = \frac{1}{S-1}\sum_{s=1}^{S}(a_s - \bar{s})^2 \tag{2.23}$$

| | |
|---|---|
| Clayton Semi-Parametric | 10463.99 |
| Clayton Parametric | 10468.29 |
| Gumbel Parametric | 10461.22 |
| Joe Parametric | 10463.39 |
| Frank Parametric | 10465.96 |

Table 2.7: WAIC for various copula survival models applied to ARIC dataset

We see that all four of the copula models perform similarly in terms of information criteria; but the Gumbel-Hougaard performs the best. In Figure 2.21 and 2.22, we display the posterior joint survivor functions for each model. Again, we can see that the posterior survival surfaces for each model are quite similar between the models.

## 2.8 Discussion and Future Work

In this paper, we have provided a systematic approach for generalizing the formation of copula models to include non-proportional marginal regressions to provide a flexible joint survival function. This approach is promising because it de-restricts the effect of the covariate on the survival functions in order to provide a more flexible and data-driven model fit. Furthermore, this approach generalizes pre-existing parametric approaches that assume proportional hazards models in the margins. Our approach performed well in simulation studies of moderate size, exhibiting low bias for all model parameters. We extended the model to include a non-parametric spline basis formulation of the baseline hazard. Using a B-spline basis increased flexibility in the model and penalizing the coefficients resulted in excellent recovery of model parameters. Such penalization is necessary to control the control points of the B-Spline basis functions that lie outside of the time domain. We obtained reasonable posterior estimates for diffuse priors, suggesting no pathologies in the underlying modeling framework. More work needs to be performed, however, in the determination of the

| Copula | Param. | Mean | SD | 2.5th | 97.5th | Copula | Param. | Mean | SD | 2.5th | 97.5th |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Clayton | $\beta_{11}$ | 0.718 | 0.098 | 0.532 | 0.918 | Gumbel | $\beta_{11}$ | 0.706 | 0.095 | 0.512 | 0.907 |
| | $\beta_{12}$ | 0.264 | 0.149 | -0.040 | 0.541 | | $\beta_{12}$ | 0.287 | 0.150 | -0.013 | 0.576 |
| | $\beta_{13}$ | 0.054 | 0.008 | 0.037 | 0.071 | | $\beta_{13}$ | 0.324 | 0.008 | 0.230 | 0.420 |
| | $\beta_{14}$ | 0.171 | 0.064 | 0.045 | 0.294 | | $\beta_{14}$ | 0.133 | 0.063 | 0.004 | 0.260 |
| | $\beta_{15}$ | 0.006 | 0.002 | 0.002 | 0.011 | | $\beta_{15}$ | 0.158 | 0.002 | 0.047 | 0.271 |
| | $\beta_{16}$ | 0.507 | 0.121 | 0.264 | 0.739 | | $\beta_{16}$ | 0.499 | 0.119 | 0.266 | 0.733 |
| | $\beta_{17}$ | -0.098 | 0.142 | -0.377 | 0.182 | | $\beta_{17}$ | -0.115 | 0.144 | -0.407 | 0.165 |
| | $\beta_{21}$ | 0.058 | 0.111 | -0.171 | 0.269 | | $\beta_{21}$ | 0.066 | 0.108 | -0.152 | 0.298 |
| | $\beta_{22}$ | 0.231 | 0.168 | -0.090 | 0.558 | | $\beta_{22}$ | 0.254 | 0.161 | -0.082 | 0.586 |
| | $\beta_{23}$ | 0.082 | 0.010 | 0.061 | 0.103 | | $\beta_{23}$ | 0.503 | 0.010 | 0.383 | 0.637 |
| | $\beta_{24}$ | 0.290 | 0.072 | 0.158 | 0.437 | | $\beta_{24}$ | 0.261 | 0.070 | 0.120 | 0.409 |
| | $\beta_{25}$ | 0.009 | 0.002 | 0.004 | 0.015 | | $\beta_{25}$ | 0.238 | 0.002 | 0.112 | 0.376 |
| | $\beta_{26}$ | 0.545 | 0.147 | 0.255 | 0.836 | | $\beta_{26}$ | 0.574 | 0.145 | 0.281 | 0.890 |
| | $\beta_{27}$ | -0.115 | 0.152 | -0.429 | 0.165 | | $\beta_{27}$ | -0.134 | 0.148 | -0.445 | 0.172 |
| | $\alpha$ | 0.873 | 0.202 | 0.481 | 1.270 | | $\alpha$ | 0.069 | 0.016 | 0.038 | 0.104 |
| | $\lambda_1$ | 0.000 | 0.000 | 0.000 | 0.000 | | $\lambda_1$ | 0.004 | 0.000 | 0.003 | 0.005 |
| | $\lambda_2$ | 0.000 | 0.000 | 0.000 | 0.000 | | $\lambda_2$ | 0.004 | 0.000 | 0.003 | 0.006 |
| | $\rho_1$ | 1.225 | 0.056 | 1.115 | 1.332 | | $\rho_1$ | 1.219 | 0.054 | 1.115 | 1.334 |
| | $\rho_2$ | 1.308 | 0.079 | 1.157 | 1.463 | | $\rho_2$ | 1.352 | 0.080 | 1.195 | 1.534 |
| | $r_1$ | 0.523 | 0.454 | 0.000 | 1.430 | | $r_1$ | 0.607 | 0.452 | 0.021 | 1.910 |
| | $r_2$ | 2.159 | 1.141 | 0.002 | 4.214 | | $r_2$ | 2.873 | 1.164 | 0.649 | 5.640 |
| Joe | $\beta_{11}$ | 0.700 | 0.096 | 0.509 | 0.884 | Frank | $\beta_{11}$ | 0.716 | 0.097 | 0.521 | 0.901 |
| | $\beta_{12}$ | 0.271 | 0.152 | -0.012 | 0.584 | | $\beta_{12}$ | 0.267 | 0.151 | -0.0167 | 0.572 |
| | $\beta_{13}$ | 0.053 | 0.008 | 0.036 | 0.069 | | $\beta_{13}$ | 0.054 | 0.008 | 0.0383 | 0.071 |
| | $\beta_{14}$ | 0.143 | 0.063 | 0.009 | 0.260 | | $\beta_{14}$ | 0.169 | 0.064 | 0.0449 | 0.295 |
| | $\beta_{15}$ | 0.006 | 0.002 | 0.002 | 0.011 | | $\beta_{15}$ | 0.006 | 0.002 | 0.001 | 0.011 |
| | $\beta_{16}$ | 0.489 | 0.119 | 0.268 | 0.725 | | $\beta_{16}$ | 0.509 | 0.117 | 0.268 | 0.725 |
| | $\beta_{17}$ | -0.114 | 0.148 | -0.413 | 0.168 | | $\beta_{17}$ | 0.102 | 0.148 | -0.390 | 0.197 |
| | $\beta_{21}$ | 0.047 | 0.107 | -0.172 | 0.249 | | $\beta_{21}$ | 0.060 | 0.110 | -0.164 | 0.272 |
| | $\beta_{22}$ | 0.235 | 0.160 | -0.076 | 0.548 | | $\beta_{22}$ | 0.233 | 0.164 | -0.081 | 0.563 |
| | $\beta_{23}$ | 0.080 | 0.010 | 0.059 | 0.099 | | $\beta_{23}$ | 0.082 | 0.010 | 0.061 | 0.102 |
| | $\beta_{24}$ | 0.260 | 0.071 | 0.116 | 0.395 | | $\beta_{24}$ | 0.290 | 0.071 | 0.156 | 0.439 |
| | $\beta_{25}$ | 0.009 | 0.002 | 0.004 | 0.014 | | $\beta_{25}$ | 0.009 | 0.002 | 0.004 | 0.015 |
| | $\beta_{26}$ | 0.531 | 0.142 | 0.252 | 0.810 | | $\beta_{26}$ | 0.547 | 0.145 | 0.267 | 0.839 |
| | $\beta_{27}$ | -0.129 | 0.152 | -0.424 | 0.172 | | $\beta_{27}$ | 0.115 | 0.153 | -0.416 | 0.179 |
| | $\alpha$ | 1.077 | 0.019 | 0.0418 | 0.118 | | $\alpha$ | 1.697 | 0.326 | 1.056 | 2.338 |
| | $\lambda_1$ | 0.000 | 0.000 | 0.000 | 0.000 | | $\lambda_1$ | 0.000 | 0.000 | 0.000 | 0.000 |
| | $\lambda_2$ | 0.000 | 0.000 | 0.000 | 0.000 | | $\lambda_2$ | 0.000 | 0.000 | 0.000 | 0.000 |
| | $\rho_1$ | 1.208 | 0.053 | 1.105 | 1.310 | | $\rho_1$ | 1.226 | 0.056 | 1.117 | 1.336 |
| | $\rho_2$ | 1.302 | 0.079 | 1.142 | 1.444 | | $\rho_2$ | 1.308 | 0.081 | 1.159 | 1.473 |
| | $r_1$ | 0.522 | 0.448 | 0.000 | 1.426 | | $r_1$ | 0.529 | 0.456 | 0.000 | 1.431 |
| | $r_2$ | 2.107 | 1.134 | 0.071 | 4.202 | | $r_2$ | 2.180 | 1.141 | 0.035 | 4.202 |

Table 2.8: Results of Copula Fits to ARIC dataset

| Parameter | Mean | SD | 2.5th | 97.5th |
|---|---|---|---|---|
| $\beta_{11}$ | 0.801 | 0.121 | 0.578 | 1.054 |
| $\beta_{12}$ | 0.327 | 0.169 | 0.007 | 0.662 |
| $\beta_{13}$ | 0.369 | 0.057 | 0.263 | 0.487 |
| $\beta_{14}$ | 0.170 | 0.070 | 0.033 | 0.308 |
| $\beta_{15}$ | 0.177 | 0.064 | 0.057 | 0.309 |
| $\beta_{16}$ | 0.562 | 0.136 | 0.298 | 0.835 |
| $\beta_{17}$ | -0.066 | 0.157 | -0.377 | 0.235 |
| $\beta_{21}$ | 0.158 | 0.142 | -0.109 | 0.447 |
| $\beta_{22}$ | 0.266 | 0.208 | -0.146 | 0.666 |
| $\beta_{23}$ | 0.627 | 0.087 | 0.467 | 0.808 |
| $\beta_{24}$ | 0.348 | 0.091 | 0.175 | 0.533 |
| $\beta_{25}$ | 0.306 | 0.083 | 0.151 | 0.478 |
| $\beta_{26}$ | 0.765 | 0.205 | 0.392 | 1.195 |
| $\beta_{27}$ | -0.081 | 0.195 | -0.465 | 0.307 |
| $\alpha$ | 0.884 | 0.200 | 0.517 | 1.297 |
| $b_1$ | 0.007 | 0.002 | 0.003 | 0.012 |
| $\alpha_{1,0}$ | 0.007 | 0.008 | 0.000 | 0.029 |
| $\alpha_{1,1}$ | 0.005 | 0.002 | 0.002 | 0.010 |
| $\alpha_{1,2}$ | 0.004 | 0.002 | 0.001 | 0.008 |
| $\alpha_{1,3}$ | 0.005 | 0.002 | 0.002 | 0.009 |
| $\alpha_{1,4}$ | 0.002 | 0.001 | 0.000 | 0.005 |
| $\alpha_{1,5}$ | 0.005 | 0.002 | 0.002 | 0.009 |
| $\alpha_{1,6}$ | 0.006 | 0.002 | 0.003 | 0.010 |
| $\alpha_{1,7}$ | 0.004 | 0.002 | 0.001 | 0.008 |
| $\alpha_{1,8}$ | 0.008 | 0.002 | 0.004 | 0.013 |
| $\alpha_{1,9}$ | 0.004 | 0.002 | 0.000 | 0.009 |
| $\alpha_{1,10}$ | 0.011 | 0.003 | 0.006 | 0.017 |
| $\alpha_{1,11}$ | 0.005 | 0.003 | 0.001 | 0.010 |
| $\alpha_{1,12}$ | 0.011 | 0.003 | 0.005 | 0.018 |
| $\alpha_{1,13}$ | 0.006 | 0.003 | 0.001 | 0.013 |
| $\alpha_{1,14}$ | 0.012 | 0.004 | 0.004 | 0.021 |
| $\alpha_{1,15}$ | 0.010 | 0.006 | 0.001 | 0.023 |
| $\alpha_{1,16}$ | 0.013 | 0.008 | 0.001 | 0.032 |
| $\alpha_{1,17}$ | 0.014 | 0.012 | 0.001 | 0.045 |
| $\alpha_{1,18}$ | 0.057 | 1.750 | 0.000 | 0.155 |
| $r_1$ | 1.703 | 1.113 | 0.316 | 4.470 |
| $b_2$ | 0.012 | 0.005 | 0.005 | 0.023 |
| $\alpha_{2,0}$ | 0.009 | 0.010 | 0.000 | 0.035 |
| $\alpha_{2,1}$ | 0.003 | 0.001 | 0.001 | 0.006 |
| $\alpha_{2,2}$ | 0.005 | 0.002 | 0.001 | 0.009 |
| $\alpha_{2,3}$ | 0.006 | 0.002 | 0.002 | 0.010 |
| $\alpha_{2,4}$ | 0.008 | 0.002 | 0.004 | 0.013 |
| $\alpha_{2,5}$ | 0.007 | 0.003 | 0.002 | 0.013 |
| $\alpha_{2,6}$ | 0.018 | 0.005 | 0.010 | 0.029 |
| $\alpha_{2,7}$ | 0.013 | 0.005 | 0.005 | 0.024 |
| $\alpha_{2,8}$ | 0.012 | 0.006 | 0.003 | 0.026 |
| $\alpha_{2,9}$ | 0.016 | 0.010 | 0.001 | 0.040 |
| $\alpha_{2,10}$ | 0.032 | 0.015 | 0.009 | 0.067 |
| $\alpha_{2,11}$ | 0.028 | 0.014 | 0.008 | 0.062 |
| $\alpha_{2,12}$ | 0.008 | 0.008 | 0.000 | 0.031 |
| $\alpha_{2,13}$ | 0.069 | 0.038 | 0.017 | 0.161 |
| $\alpha_{2,14}$ | 0.016 | 0.019 | 0.000 | 0.067 |
| $\alpha_{2,15}$ | 0.014 | 0.016 | 0.000 | 0.056 |
| $\alpha_{2,16}$ | 0.015 | 0.023 | 0.000 | 0.077 |
| $\alpha_{2,17}$ | 0.027 | 0.073 | 0.000 | 0.158 |
| $\alpha_{2,18}$ | 0.055 | 0.424 | 0.000 | 0.296 |
| $r_2$ | 6.550 | 2.132 | 2.679 | 11.058 |

Table 2.9: Results of Semiparametric Copula Fits to MI dataset

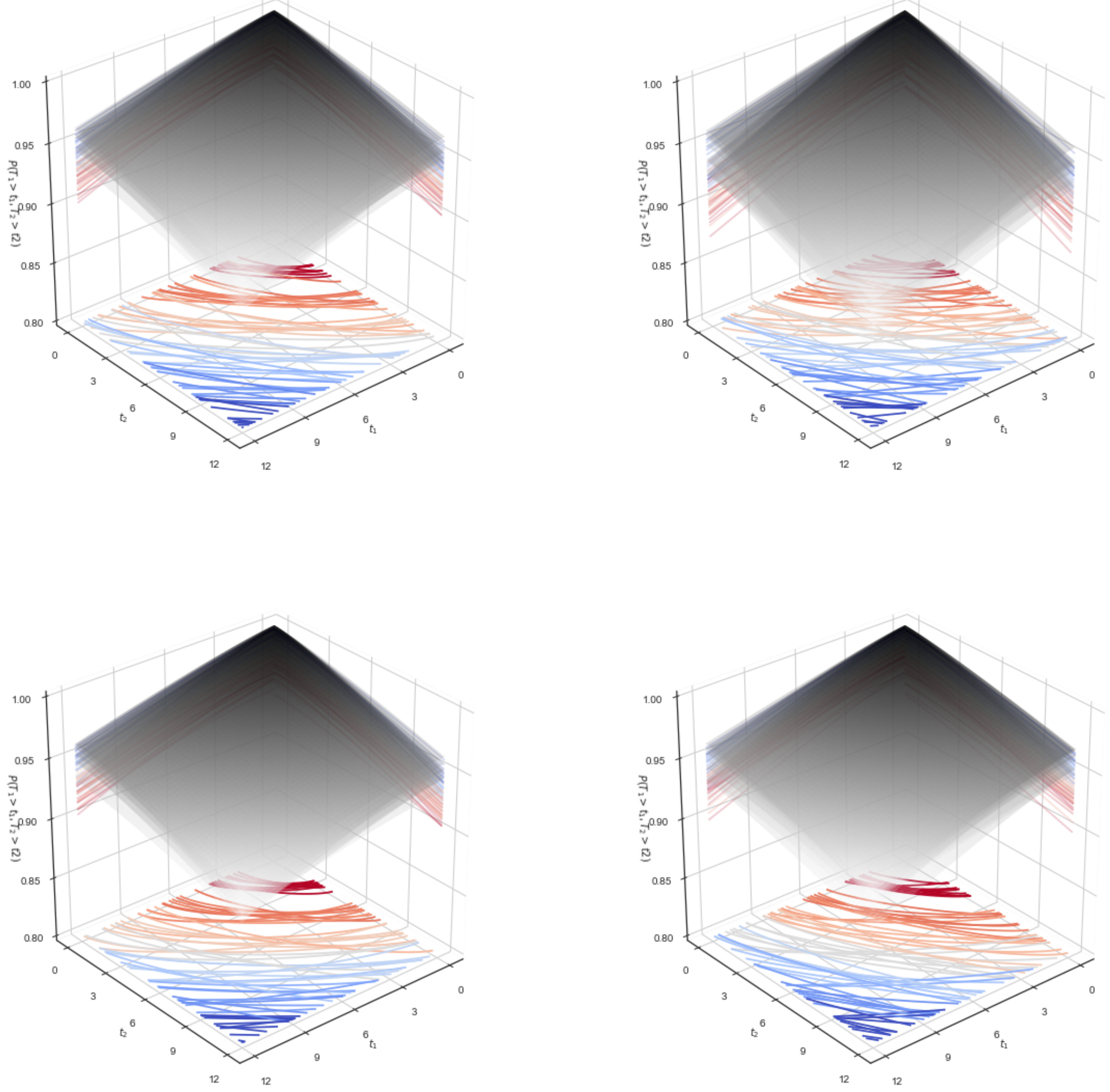


Figure 2.21: Posterior Survival Surface Draws for Female Smokers. Top left: Clayton model. Top right: Gumbel model, Bottom left: Joe Model, Bottom Right: Frank Model

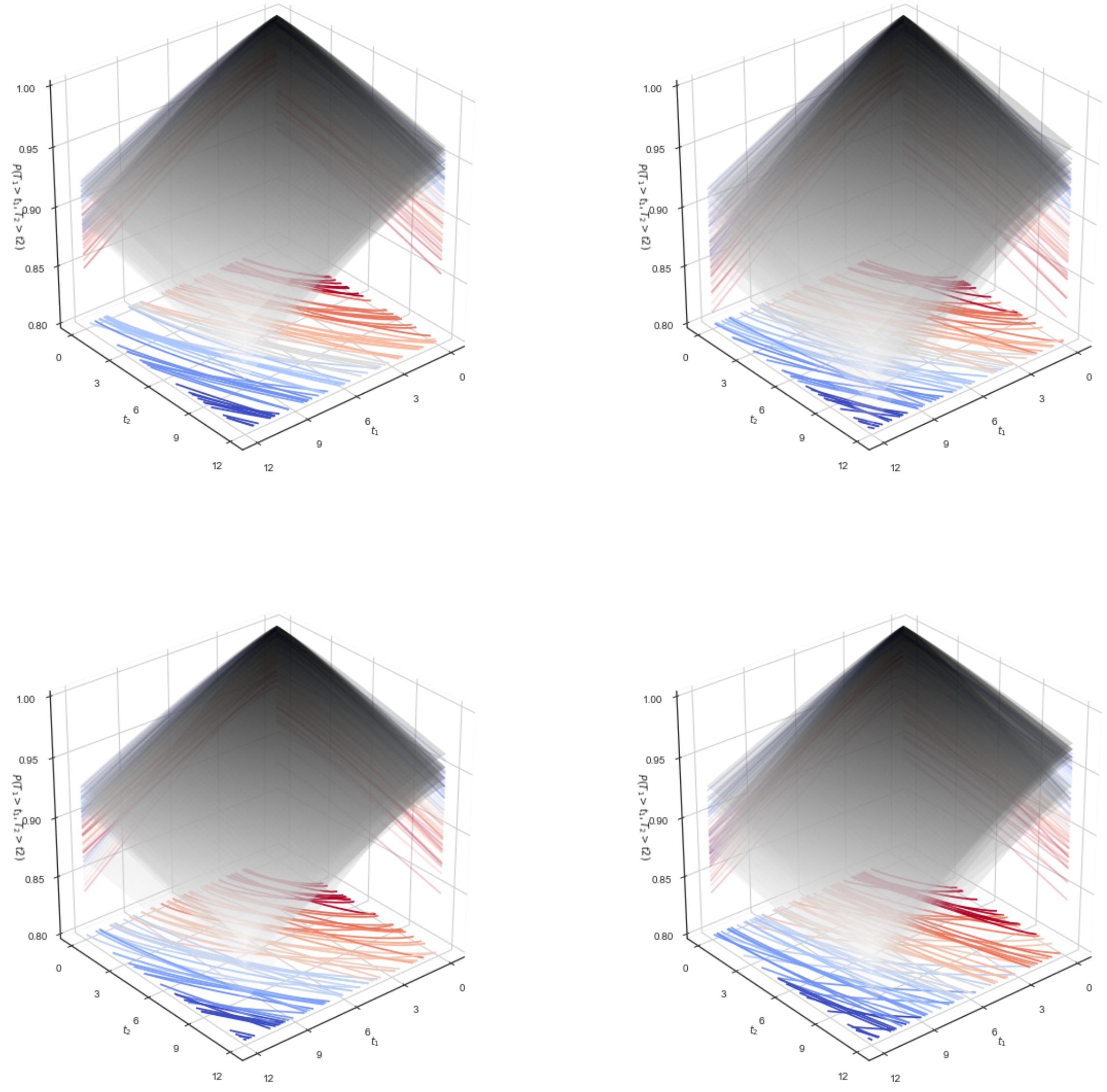


Figure 2.22: Posterior Survival Surface Draws for Male Smokers. Top right: Gumbel model, Bottom left: Joe Model, Bottom Right: Frank Model

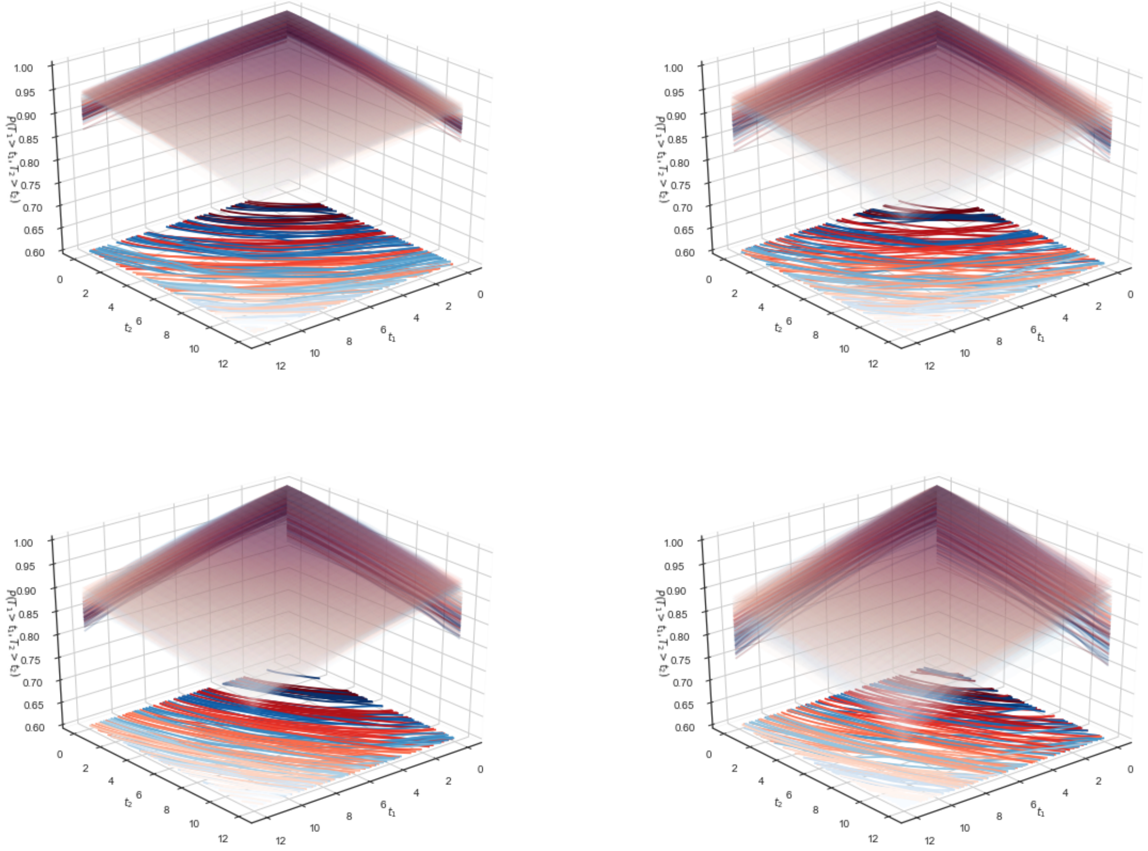


Figure 2.23: Comparison of posterior draws from joint survival function for ARIC dataset; semi-parametric Clayton model (blue), parametric Clayton model (red). Surfaces represent draws from the posterior of the joint probability of MI and stroke for age 70 with systolic blood pressure of 120. Top left panel represents female non-smokers, top right panel represents female smokers, bottom left panel represents male non-smokers, bottom right panel represents male smokers

proper knot number and placements for differing censoring rates and sample sizes; in this paper, we simply placed knots at evenly spaced sample quantiles. We applied the methodology to a myocardial infarction dataset where we fit the parametric copula models and subsequently compared to the semi-parametric model.

One main conclusion from the present paper is that we can define a robust and flexible framework that not only generalizes the survival function specification, but also the baseline hazard, resulting in a versatile tool for statistical practitioners. With a tractable, stable estimation framework such as this, we can define even more flexible models that generalize the relationship between input variables and the survival model. One simple extension of this would be the incorporation of non-linear predictor effects in the form of splined covariates. Similarly, if the specification of knots is difficult, one may alternatively specify a Gaussian process form on the covariate effects (although at the expense of $O(n^3)$ computational complexity) (Fernández et al., 2016). Through the inclusion of non-linear covariates, we envision the incorporation of image or other complex covariate types via a neural network architecture embedded within the flexible framework developed in the present paper.

We showed that the semi-parametric model worked well in moderately sized data sets, however, it is important to study large sample Bayesian models. Variational Bayesian methods (Blei et al., 2017) comprise approximate methods to overcome the computational complexity of MCMC sampling in large datasets by approximating the posterior distribution of the model parameters with a member of a parametric family closest in terms of Kullback-Leibler divergence. Such methods can be adopted to the copula framework presented in this paper to better facilitate the adoption of more complex models to larger datasets.

In the semi-parametric portion of this paper, we limited our discussion to only the Clayton copula model. In principle, however, the described B-spline methodology may be applied to any valid copula specification. However, an important consideration in the usage of copula models is the computational tractability of each model. We note that the Clayton copula likelihood has a log likelihood free of additional exponential functions whereas other Archimedean models contain more complex mathematical expressions thereby facilitating much simpler computational execution. Indeed, during the simulations, we noted a sharp increase in the computational time required to complete the MCMC from the Clayton and other listed models.

Lastly, further work needs to be done in this investigation with respect to differing censoring mechanisms. Specifically, one possible extension is to generalize the model further to incorporate general censoring structures. Left censoring is straight forward (see Appendix), however, the general censoring case is less straightforward and requires further research.

## 2.9 Appendix

### 2.9.1 Derivatives of Copulas for Likelihood Specification

#### *2.9.1.1 Gumbel Model*

The joint survival function for the Gumbel model is given by

$$C_\alpha(u_1,u_2) = e^{-((-\log(u_2))^\alpha - \log^\alpha(u_1))^{\frac{1}{\alpha}}} \tag{2.24}$$

The respective derivatives of the copula terms with respect to each argument are

$$\frac{\partial C_\alpha(u_1,u_2)}{\partial u_1} = \frac{1}{u_1}\left((-\log(u_2))^\alpha - \log^\alpha(u_1)\right)^{\frac{1}{\alpha}(-\alpha+1)} e^{-((-\log(u_2))^\alpha - \log^\alpha(u_1))^{\frac{1}{\alpha}}} \log^{\alpha-1}(u_1) \tag{2.25}$$

and

$$\frac{\partial C_\alpha(u_1,u_2)}{\partial u_2} = -\frac{(-\log(u_2))^\alpha}{u_2\log(u_2)}\left((-\log(u_2))^\alpha - \log^\alpha(u_1)\right)^{-\frac{1}{\alpha}(\alpha-1)} e^{-((-\log(u_2))^\alpha - \log^\alpha(u_1))^{\frac{1}{\alpha}}} \tag{2.26}$$

Furthermore, the corresponding joint density of the copula model is given by

$$\begin{aligned} c_\alpha(u_1,u_2) &= \frac{\partial^2 C_\alpha(u_1,u_2)}{\partial u_1 \partial u_2} \\ &= \frac{(-\log(u_2))^\alpha \log^{\alpha-1}(u_1)}{u_1 u_2 \log(u_2)} \\ &\quad \times \left((-\log(u_2))^\alpha - \log^\alpha(u_1)\right)^{-2+\frac{1}{\alpha}} \left(-\alpha - \left((-\log(u_2))^\alpha - \log^\alpha(u_1)\right)^{\frac{1}{\alpha}} + 1\right) \\ &\quad \times e^{-((-\log(u_2))^\alpha - \log^\alpha(u_1))^{\frac{1}{\alpha}}} \end{aligned}$$

#### *2.9.1.2 Joe Model*

$$\begin{aligned}\frac{\partial C_\alpha(u_1,u_2)}{\partial u_1} &= \left(-u_1+1\right)^{\alpha-1}\left(-\left(-u_2+1\right)^{\alpha}+1\right)\\ &\quad\times\left(-\left(-u_1+1\right)^{\alpha}\left(-u_2+1\right)^{\alpha}+\left(-u_1+1\right)^{\alpha}+\left(-u_2+1\right)^{\alpha}\right)^{\frac{1}{\alpha}(-\alpha+1)}\end{aligned}$$

$$\begin{aligned}\frac{\partial C_\alpha(u_1,u_2)}{\partial u_2} &= \left(-u_2+1\right)^{\alpha-1}\left(-\left(-u_1+1\right)^{\alpha}+1\right)\\ &\quad\times\left(-\left(-u_1+1\right)^{\alpha}\left(-u_2+1\right)^{\alpha}+\left(-u_1+1\right)^{\alpha}+\left(-u_2+1\right)^{\alpha}\right)^{\frac{1}{\alpha}(-\alpha+1)}\end{aligned}$$

$$\begin{aligned}c_\alpha(u_1,u_2) &= \frac{\partial^2 C_\alpha(u_1,u_2)}{\partial u_1 \partial u_2}\\ &= \left(-u_1+1\right)^{\alpha-1}\left(-u_2+1\right)^{\alpha-1}\\ &\quad\times\left(-\left(-u_1+1\right)^{\alpha}\left(-u_2+1\right)^{\alpha}+\left(-u_1+1\right)^{\alpha}+\left(-u_2+1\right)^{\alpha}\right)^{-2+\frac{1}{\alpha}}\\ &\quad\times\left(\alpha\left(\left(-u_1+1\right)^{\alpha}-1\right)\left(\left(-u_2+1\right)^{\alpha}-1\right)+\alpha\left(-\left(-u_1+1\right)^{\alpha}\left(-u_2+1\right)^{\alpha}+\left(-u_1+1\right)^{\alpha}\right.\right.\end{aligned}$$

#### *2.9.1.3 Frank Model*

$$C_\alpha(u_1,u_2) = -\frac{1}{\alpha}\log\left(1+\frac{\left(e^{\alpha u_1}-1\right)\left(e^{\alpha u_2}-1\right)}{-1+e^{-\alpha}}\right) \tag{2.27}$$

$$\frac{\partial C_\alpha(u_1,u_2)}{\partial u_1} = \frac{\left(e^{\alpha u_2}-1\right)e^{\alpha(u_1+1)}}{-\left(e^{\alpha u_1}-1\right)\left(e^{\alpha u_2}-1\right)e^{\alpha}+e^{\alpha}-1}$$

$$\frac{\partial C_\alpha(u_1,u_2)}{\partial u_2} = \frac{\left(e^{\alpha u_1}-1\right)e^{\alpha(u_2+1)}}{-\left(e^{\alpha u_1}-1\right)\left(e^{\alpha u_2}-1\right)e^{\alpha}+e^{\alpha}-1}$$

$$\begin{aligned} c_\alpha(u_1,u_2) &= \frac{\partial^2 C_\alpha(u_1,u_2)}{\partial u_1 \partial u_2} \\ &= \frac{\alpha\left(e^{\alpha}-1\right)e^{\alpha(u_1+u_2+1)}}{\left(-\left(e^{\alpha u_1}-1\right)\left(e^{\alpha u_2}-1\right)e^{\alpha}+e^{\alpha}-1\right)^2} \end{aligned}$$

#### *2.9.1.4 Clayton Model*

$$\frac{\partial C_\alpha(t_1,t_2)}{\partial t_1} = t_1^{-(\alpha+1)}\left(t_1^{-\alpha}+t_2^{-\alpha}-1\right)^{-\left(\frac{\alpha+1}{\alpha}\right)}$$

$$\frac{\partial C_\alpha(t_1,t_2)}{\partial t_2} = t_2^{-(\alpha+1)}\left(t_1^{-\alpha}+t_2^{-\alpha}-1\right)^{-\left(\frac{\alpha+1}{\alpha}\right)}$$

$$S_{ik}(t|X) = \left\{1+r_k\Lambda(t)\exp\{\beta_i^{'}X_i\}\right\}^{-r_k^{-1}}$$

$$\begin{aligned} c_\alpha(t_1,t_2) &= \frac{\partial^2 C_\alpha(t_1,t_2)}{\partial t_1 \partial t_2} \\ &= (\alpha+1)(t_1t_2)^{-(\alpha+1)}(t_1^{-\alpha}+t_2^{-\alpha}-1)^{-\frac{2\alpha+1}{\alpha}} \end{aligned}$$

### 2.9.2 Simulation Details

Simulation of the preceding models comprises two steps: 1. Generation from the copula margins and 2. transformation of the uniformly distributed margins in step 1 into the transformation models.

#### *2.9.2.1 Marginal Data*

The generation of the data for the transformation survival model for the non-parametric involves a baseline hazard that comprises two joined Weibull hazards. For this specification, we define the parametric parameters, $\lambda_1, \lambda_2.\rho_1, \rho_2$ and a change-point, $\kappa$. Under this parameterization, we generate data from the model with baseline hazard

$$\lambda_0(t) = \lambda_1 \rho_1 t^{\rho_1 - 1} I(t < \kappa) + \lambda_2 \rho_2 t^{\rho_2 - 1} I(t \geq \kappa) \tag{2.28}$$

With this formulation, we can define the baseline cumulative hazard function, which is necessary for use in the inverse probability transform:

$$\begin{aligned} \Lambda_0(t) &= \int_0^t \lambda_0(t) dt \\ &= \lambda_1 t^{\rho_1} + \left(\lambda_2 t^{\rho_2} - \lambda_1 \kappa^{\rho_1}\right) I(t \geq \kappa) \end{aligned}$$

For simulation, we need the inverse function of $\Lambda_0(t)$, which is defined as

$$\Lambda_0^{-1}(u) = \left(\frac{u}{\lambda_1}\right)^{1/\rho_1} I(u \leq \Lambda_0(\kappa)) + \left(\frac{u - \lambda_1 \kappa^{\rho_1} + \lambda_2 \kappa^{\rho_2}}{\lambda_2}\right)^{1/\rho_2} I(u > \Lambda_0(\kappa))$$

Finally, we invert the remainder of the transformation model. First, with uniform variates $U$, we simulate from the conditional cumulative hazard by

$$T = \Lambda_0^{-1}\left(\frac{U^{-r} - 1}{r \exp(X'\beta)}\right)$$

### 2.9.3 Technical Considerations for Likelihood

Here, we provide some technical considerations for formulating the censored likelihood derived above. Further notes can be found in Georges et al. (2001). Here, we consider the joint CDF of the observed random variables $T_1^* = \min(T_1, C_1)$ and $T_2^* = \min(T_2, C_2)$ such that $T_i$ have distribution functions $F_{T_i}(t_i)$ and censoring mechanisms such that $C_i$ have distribution functions $F_{C_i}(c_i)$.

First, we can see in the case that neither

$$\begin{aligned}
P(T_1^* \leq d_1, T_2^* \leq d_2) &= P(C_1 \leq d_1, T_2^* \leq d_2, T_1 > C_1) \\
&= \int\int\int 1_{[c<d_1, t_2 \leq d_2, t_1 > c]} f(t_1, t_2) f_{C_1}(c) dt_1 dt_2 dc \\
&= \int_0^{d_1} \left[ \int_c^{\infty} \int_0^{d_2} f(t_1, t_2) \right] f_{C_1}(c) dc \\
&= \int_0^{d_1} \left[ S_1(c) - C_\alpha(S_1(c), S_2(d_2)) \right] f_{C_1}(c) dc
\end{aligned}$$

Using Leibniz's rule and integration by parts, we obtain a likelihood contribution of

$$L \propto \partial C_\alpha(S_1(c), S_2(d_2))$$

Theoretically, the extension to the left censored likelihood could proceed with the preceding derivation via

$$\begin{aligned}
P(T_1^* \leq d_1, T_2^* \leq d_2) &= P(C_1^- \leq d_1, T_2^* \leq d_2, T_1 \leq C_1^-) \\
&= \int\int\int 1_{[c \leq d_1, t_2 \leq d_2, t_1 \leq c]} f(t_1, t_2) f_{C_1^-}(c) dt_1 dt_2 dc \\
&= \int_0^{d_1} \left[ \int_0^{c} \int_0^{d_2} f(t_1, t_2) \right] f_{C_1^-}(c) dc \\
&= \int_0^{d_1} \left[ 1 - S_1(c) - S_2(c) + C_\alpha(S_1(c), S_2(d_2)) \right] f_{C_1}(c) dc
\end{aligned}$$

#### *2.9.3.1 Accuracy and Speed of Spline integral implementation*

Consider the function $f(x) = .5x + .1x^2 + 3$ and suppose we are interested in the integral $f(t) = \int_0^t f(x) dx$. In Figure 2.24 we see that the function can be represented by a spline function with B-spline basis functions.

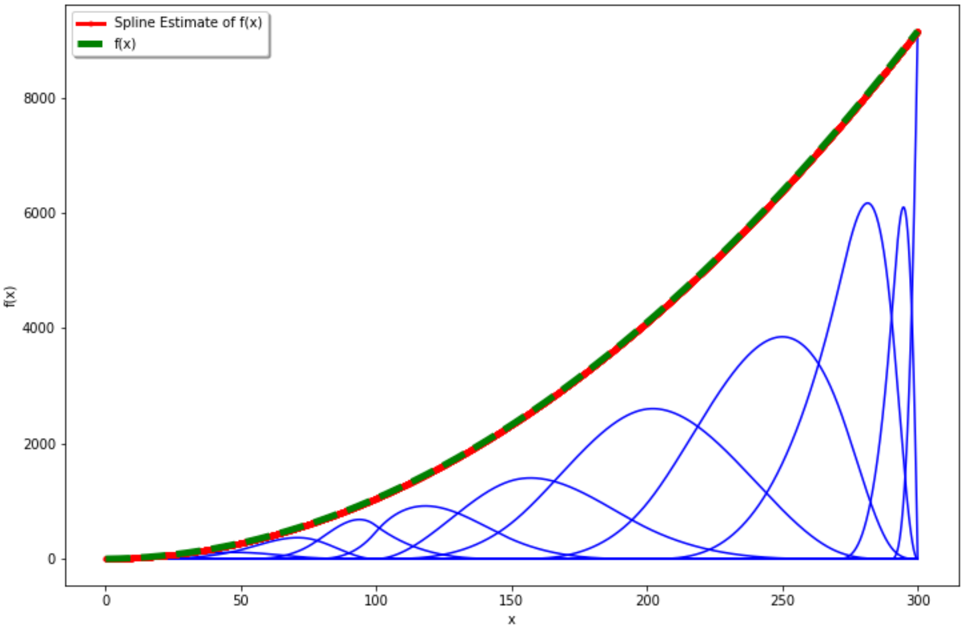


Figure 2.24: True function $f(x)$ represented by a spline

Now, we compare three four possible methods of integration. As a benchmark, we calculate the resulting value from the method and its time to completion. We repeat this process 100 times and report the mean and standard deviation of both the time and the evaluated integral value at $t = 30$ (which equals 1215.0). Table 2.10 contains the results of this simulation.

1. Quadrature integration of $f(x)$

2. Quadrature integration of $f(\hat{x})$

3. Looped Integral Formula for $\int_0^t f(\hat{x})dx$

4. Matrix Optimized integration routine (our approach)

We can see that the matrix-optimized version of the spline gives an extremely accurate value for the required integration. Of course, integrating the actual function is fastest, but in terms of evaluating the integral of the spline representation,

|  | Mean | SD Value | Mean Time | SD Time |
|---|---|---|---|---|
| Quadrature True Function | 1215.0 | $< 0.001$ | $< 0.001$ | $< 0.001$ |
| Quadrature Spline Function | 1215.0 | $< 0.001$ | 0.1100 | 0.0129 |
| Loop Spline | 1215.0 | $< 0.001$ | 0.0802 | 0.0097 |
| Matrix Optimized | 1215.0 | $< 0.001$ | 0.0059 | 0.0011 |

Table 2.10: Results of spline integration simulation

our matrix-optimized version is superior to the quadrature integration and the pre-existing looped integral.

### 2.9.4 Alternative initialization scheme results

In table 2.11, we present the results of a two-stage initialization procedure for MCMC sampling. We notice that the coverage of the 95 percent credible intervals are between 93.2 and 96.6 at the nominal level. The secondary initialization routine first estimates the marginal parameters, then univariately optimizes the association parameter conditional on the estimated marginal model. The greatest difference between the results presented in the first initialization step and the initialization presented here is a reduction in the bias of posterior mean for the copula association parameter which gave a reduction of 0.0114. This indicates that the joint optimization initialization routine did not artificially select “good” datasets that may have falsely informed the posterior coverage probabilities in the first round of simulations.

| Parameter | Truth | ABPM<br>(×100) | SDPM<br>(×100) | SDS<br>(×100) | MSE<br>(×100) | CP<br>(×100) |
|---|---|---|---|---|---|---|
| $\beta_{11}$ | 1.63 | 0.157 | 4.397 | 4.605 | 0.194 | 95.2 |
| $\beta_{12}$ | 0.03 | -0.009 | 0.171 | 0.167 | $< 0.001$ | 94.4 |
| $\beta_{21}$ | 0.8 | -0.274 | 3.648 | 3.712 | 0.134 | 95.4 |
| $\beta_{22}$ | 0.03 | -0.004 | 0.175 | 0.169 | $< 0.001$ | 93.2 |
| $\alpha$ | 3.21 | 0.148 | 8.461 | 8.144 | 0.716 | 95.4 |
| $\lambda_1$ | 0.0047 | 0.004 | 0.051 | 0.051 | $< 0.001$ | 94.8 |
| $\lambda_2$ | 0.0037 | 0.104 | 0.051 | 0.051 | $< 0.001$ | 96.6 |
| $\rho_1$ | 0.716 | 0.104 | 1.390 | 1.378 | 0.019 | 95.0 |
| $\rho_2$ | 0.725 | 0.062 | 1.373 | 1.439 | 0.019 | 96.2 |
| $r_1$ | 0.5 | 0.253 | 3.622 | 3.665 | 0.132 | 94.4 |
| $r_2$ | 0.5 | 0.265 | 3.707 | 3.967 | 0.138 | 96.4 |

Table 2.11: Copula simulation results for 500 simulations for parametric copulas, second initialization scheme

CHAPTER 3

# BAYESIAN LARGE SCALE INFERENCE FOR TIME TO FIRST EVENT OF MULTIVARIATE ORDINAL SURVIVAL OUTCOMES WITH APPLICATION TO AN OBSERVATIONAL COHORT STUDY

## 3.1 Abstract

Motivated by a large scale observational cohort study, we develop a scalable Bayesian framework to accommodate time to first event of multivariate survival outcomes with ordinal severity. We model the multivariate survival outcomes using a flexible gamma frailty transformation model that includes Cox-proportional hazards model and proportional odds model as two special cases. A computationally efficient algorithm based on variational inference is used to scale the Bayesian inferential scheme to large datasets. Bayesian model selection procedures are further developed to determine the most proper and pragmatic transformation. Numerical simulations are conducted to evaluate the validity of the method and variational algorithm. The proposed method is further applied to a cohort with 163,763 patients from the Tennessee Asthma Bronchiolitis Study (TABS).

## 3.2 Introduction

Pervasive throughout the medical sector, electronically-held medical databases contain information on the order of hundreds of thousands to millions of records from which to study. These data are collected for a variety of reasons, but can nonetheless be leveraged for biomedical research more cost effectively than conducting a randomized controlled trial. One class of medical administrative data that has found popularity among medical researchers is medical claims data (Cave and Munson, 1999). In the United States, many states with specific insurance systems maintain databases that may be accessed for use in academic research. In healthcare insurance claims data, information pertaining only to the *type* of healthcare administered is recorded since other information is superfluous for billing purposes. Furthermore, typical patient data include individual characteristics, frequency of healthcare visits, and healthcare type may be recorded in these databases. Although limited by the granularity of information in this type of encoding, the latter field is routinely used as an ordinal proxy for the severity of a particular disease. Hypothetically, one might expect a patient who was hospitalized for a condition would have experienced a more severe form of the disease than a patient who only visited an outpatient clinic. In the state of Tennessee, for example, individuals who qualify under certain income thresholds receive state-funded healthcare insurance from the Tennessee Division of Health Care Finance and Administration under the Tennessee Medicaid Program (TennCare) (Carroll et al., 2008). Consequently, personalized health information about subjects enrolled in TennCare is recorded and retained for each patient. Our motivating study, the Tennessee Asthma Bronchiolitis Study (TABS), uses TennCare data to study how maternal smoking durning pregnancy relates to infant bronchiolitis outcomes. In this dataset, there are three possible types of healthcare visitation types: outpatient clinic, emergency department, and hospitalization. As the immune response to bronchiolitis changes after repeated healthcare visits, our study focuses on time to the first of these three events.

Time-to-first event analysis is typically formulated in terms of a univariate proportional hazards model (Anker and McMurray, 2012), Rauch et al. (2018), (Claggett et al., 2018). Furthermore, these analyses are typically conducted in the recurrent events framework and study time to the first of same type of event (the first of a possible chain of recurrent verses modeling all events), whereas in our case we have time to the first of several different type of events. There is limited literature on the latter. Now, fitting a univariate model to the time-to-first event effectively assumes the same regression effect on different type of events. In the preliminary analysis of time to each type of the event (by ignoring within subject correlation) in TABS study, the maternal smoking effect from the Cox proportional model varies across different type of bronchiolitis-related healthcare admission (see Appendix for these univariate models). In this paper, we study the topic under Bayesian framework with 'big data' that allows different regression effects on different type of events with a general transformation model that includes proportional hazard model and proportional odds model as two special cases. With 'big data', we also study scaling the method to larger datasets using contemporary variational inference (Blei et al., 2017).

In the TABS study, the type of healthcare admission was considered to be associated with disease severity – patients with hospitalization for example, typically experienced higher severity of the disease than those who visited out patient facilities. Hence there is an implicit ordering to the events of interest. There is, however, limited literature on developing statistical inference for time to first event of multivariate ordinal survival outcomes. Statistical inference for simultaneously determining the effect of treatment on severity of event and time to event has its history in frequentist statistical literature. The systematic analysis of ordinal survival data was first explored by Berridge and Whitehead (1991), wherein the authors conceptualized the ordinal outcome using a continuation ratio model for time to any event and using one proportional hazards model. The two models are linked by incorporating the function of time to *any* event as a covariate in a continuation ratio sub-model. The continuation ratio component of this model has the limitation that disease severity is a function of time; rather, in our situation, we do not assume that the disease progresses in a manner that is amenable to this formulation. Similarly, Falcaro and Pickles (2007) proposed flexible mixed probit model involving person-specific thresholds for an interval censoring mechanism on the ordinal survival outcome representing censored age of disease onset. Subject specific parameter estimation is computationally difficult even in frequentist models due to the complexity of the underlying high dimensional optimization. Additionally, the authors consider the time itself as ordinal rather than the outcome of interest, as ordinal. In both Hedeker et al. (2000) and Thomas and Have (1996), the authors consider modeling the outcome a using complementary log-log transformation models. Similarly to Falcaro and Pickles (2007), the first of these texts treats the survival time itself as an ordinal outcome rather than modeling time to a disease with ordinal severity. Again, the latter of these requires the estimation of subject specific thresholds. While these texts offer important theoretical and methodological foundations to the modeling of complex multivariate survival outcomes, no authors have developed a general methodology for the simultaneous modeling of disease severity and time-to-event using proxy observations in such a way that can use validation data. Furthermore, all of the reviewed methods have been developed within the frequentist paradigm and as such, no Bayesian solution for the analysis of multivariate ordinal event data has been proposed in the literature.

To model the ordinal survival outcome flexibly, we develop and employ a specific transformation gamma frailty model (de Castro et al., 2014) and adapt it to large scale datasets using approximate Bayesian inference. Secondly, we convolve the results of the multivariate frailty model with data that correlates the ordinal outcome with disease severity to obtain an overall effect size for the covariate of interest. Instead of having to derive distributional results for a combination step, the incorporation of validation data for the ordinal proxy variable will be straightforward using the results of the Bayesian posterior computation.

One pervasive challenge in the area of Bayesian statistics, however, is the efficient and accurate computation of posterior quantities with large amounts of data. Since the introduction of the Metropolis-Hastings algorithm (Chib and Greenberg, 1995) and associated software such as BUGS (Spiegelhalter et al., 1994), Markov Chain Monte Carlo has become a mainstay in Bayesian computation. While more efficient sampling schemes such as the Hamiltonian Monte Carlo routine (Hoffman and Gelman, 2014) have drastically improved sampling performance, MCMC remains computationally slow for relatively complex models with more that a few thousand observations. Due to this, much research has been conducted addressing the problem of scalable Bayesian computation; see (Bardenet et al., 2015) and (Blei et al., 2017) for good references. Indeed, the methodological motivation behind this paper comes from the usage of large medical insurance databases to query data. As such, any proposed Bayesian model that a researcher desires to use on such data must inherently address the issue of scalability with large amounts of data. Therefore, as another focus of this paper, we describe in detail a scalable alternative to MCMC using variational inference (Blei et al., 2017). We describe the overall framework and provide tools necessary for proper implementation.

The remainder of the paper is organized in the following manner. In the next section, we describe the general formulation of the model for the time to first multivariate events as well as develop a weighted composite score to quantify the overall exposure effect by incorporating external data. In Section 3, we outline the inferential methodology used to fit the model described in Section 2. In Section 4, we present results of a simulation study to assess the validity of the approach. Finally, in Section 5, we apply the methodology to our motivating study consisting of over 160,000 patients involved in a bronchiolitis retrospective cohort study. A discussion and future directions section concludes the paper.

## 3.3 General Model Formulation

In this section, we develop the inferential framework for time to first event analysis. Specifically, we develop a generalized transformation model formulation of the gamma frailty multivariate survival analysis model (de Castro et al. (2014), Zeng et al. (2009)). At this stage, we present the model in general terms, with no explicit mathematical representation of the baseline hazard function so as to illustrate the influence of the transformation parameter. We will define explicit form for this component in the next section.

With respect to the shape of hazard function over differing values of a covariate, in general, we would like to account for non-proportionality of hazards of a disease over time. That is to say, in instances where the difference in hazards between differing groups changes as a function of time, it is clear that the traditional proportional hazards model would be inappropriate. Although other alternatives exist, in this paper we consider the transformation given by

$$G(v, r) = vI(r = 0) + [\log(1 + vr)/r]I(r > 0). \tag{3.1}$$

for $v, r \in \mathcal{R}^+$ As values of $r$ increase, the outputs of the transformation for the varying parameter are more similar for successive values of $v$ than outputs for smaller values of $r$.

Continuing with the model formulation, we introduce this particular transformation into the survival framework by conceptually inputting the conditional cumulative hazard of proportional hazards model into the aforementioned transformation $G$. Within this transformation framework, there are two cases that correspond to pre-existing survival models with interpretable effect sizes. Specifically, as $r \to 0$, the model approaches the proportional hazards model, while when $r = 1$, we recover the proportional odds survival model (Bennett, 1983). To proceed with the inferential framework, we now introduce some notation.

### 3.3.1 Notation and Basic Quantities

Although we are primarily interested in modeling a group effect across differing levels of disease severity, our data contains observations of only the first instance of

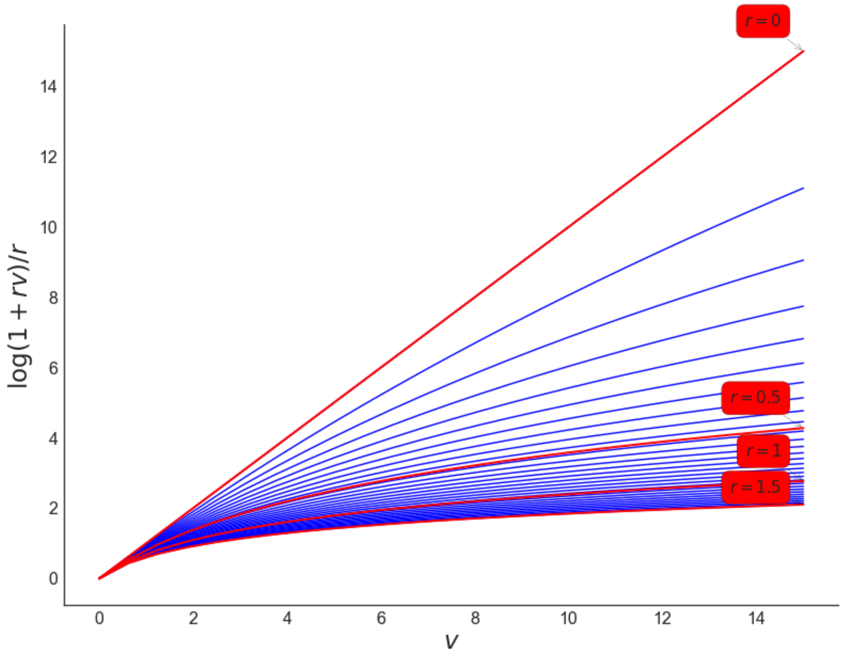


Figure 3.1: Hazard and Survival Functions subject to the Transformation $G$

the severity of disease. That is to say, we do not observe the presence (or absence) of all disease severity states. In this framework, we incorporate the unobserved states as right-censored.

Let $Y_k$ denote the true event time of the $k^{th}$ event $k = 1, \ldots, K$ and $Y_0 = C$, the censoring time. We observe $T = \min\{Y_0, Y_1, \ldots, Y_K\}$ and $\delta = \{k : Y_k \leq Y_j, \forall j \neq k\}$. To account for correlation within an individual, we specify a frailty term, denoted by $\omega_i$. We parametrically model the frailty by assuming $\omega_i \sim \Gamma(\theta^{-1}, \theta^{-1})$, to ensure identifiability. As discussed, we consider a transformed version of the conditional cumulative hazard function under $G$. Hence, we consider the following model of the conditional cumulative hazard function:

$$\begin{aligned}\Lambda_{ik}(t|\omega_i, X_i) &= \omega_i G\left[\Lambda_k(t)\exp(\beta_k^T X_i)\right] &(3.2)\\ &= \omega_i \log\left[1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right]/r. &(3.3)\end{aligned}$$

From this specification, we conclude that the survival function and hazard functions are given respectively by

$$S_{ik}(t|\omega_i, X_i) = \exp\left\{-\omega_i \log\left[1 + r\Lambda_k(t)\exp(X_i^T\boldsymbol{\beta}_k)\right]/r\right\} \tag{3.4}$$

and

$$\lambda_{ik}(t|\omega_i, X_i) = \lambda_k(t)\exp(\beta_k^T X_i)\omega_i\left\{1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right\}^{-1}. \tag{3.5}$$

### 3.3.2 Likelihood Formulation

Given the transformation model in equation (3.3), we now construct the likelihood associated with this formulation. As previously stated, if a particular level of disease severity is observed for an individual, then the other $K-1$ levels are considered right censored at the time that event was observed. Hence for each of these $K-1$ levels, we introduce the respective survival function of that level. Specifically, we specify the conditional joint probability distribution of the event time and $\delta_i$ by

$$\begin{aligned} P(T_i = t, \delta_i = k|\omega_i, X_i, r>0) &= \prod_{k=1}^{K}\left[f_k(t)\prod_{\{j:j\neq k, j=1,\ldots,K\}} S_j(t)\right]^{I(\delta=k)}\left[S_k(t)\right]^{I(\delta=0)} \\ &= \prod_{k=1}^{K}\left\{\frac{\lambda_k(t)\exp(\beta_k^T X_i)\omega_i}{1+r\Lambda_k(t)\exp(\beta_k^T X_i)}\right\}^{I(\delta=k)} \\ &\qquad \times\exp\left\{-\omega_i\log\left[1+r\Lambda_k(t)\exp(\beta_k^T X_i)\right]/r\right\} \end{aligned}$$

We note that the frailties, $\omega_i$, are nuisance parameters and we integrate them with respect to their density such that for $\boldsymbol{\theta} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K, \lambda_1, \ldots, \lambda_K, \theta\}$,

$$\begin{aligned} L(\boldsymbol{\theta}|X, r>0) &= \prod_{i=1}^{n}\int_0^\infty P(T=t, \delta_i = k|\omega_i, X_i)P(\omega_i|\theta)d\omega_i \\ &= \prod_{i=1}^{n}\frac{\Gamma\left(\sum_{k=1}^K I(\delta_i=k)+\theta^{-1}\right)\theta^{-\theta^{-1}}\prod_{k=1}^K\left[\frac{\lambda_k(t)\exp(\beta_k^T X_i)}{1+r\Lambda_k(t)\exp(\beta_k^T X_i)}\right]^{I(\delta_i=k)}}{\left(\theta^{-1}+\sum_{k=1}^K\log\left(1+r\Lambda_k(t)\exp(\beta_k^T X_i)\right)/r\right)^{\sum_{k=1}^K I(\delta_i=k)+\theta^{-1}}\Gamma(\theta^{-1})}. \end{aligned}$$

Furthermore, because $\frac{\Gamma(\alpha+1)}{\Gamma(\alpha)} = \alpha$ and since $\sum_{k=1}^{K} I(\delta_i = k) \in \{0, 1\}$, we conclude that

$$L(\boldsymbol{\theta}|X, r > 0) \;=\; \prod_{i=1}^{n} \frac{\theta^{-\sum_{k=1}^{K} I(\delta_i=k)} \theta^{-\theta^{-1}} \prod_{k=1}^{K} \left[\frac{\lambda_k(t)\exp(\beta_k^T X_i)}{1+r\Lambda_k(t)\exp(\beta_k^T X_i)}\right]^{I(\delta_i=k)}}{\left(\theta^{-1} + \sum_{k=1}^{K} \log\left(1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right)/r\right)^{\sum_{k=1}^{K} I(\delta_i=k)+\theta^{-1}}} \tag{3.6}$$

Similarly, for the case where $r = 0$ we have the joint probability of

$$\begin{aligned} P\left(T_i = t, \delta_i = k|\omega_i, X_i, r = 0\right) \;&=\; \prod_{k=1}^{K} \left[\omega_i \lambda_k(t)\exp(\beta_k^T X_i)\right]^{I(\delta=k)} \qquad (3.7) \\ &\quad \times \exp\left\{-\omega_i \Lambda_k(t)\exp(\beta_k^T X_i)\right\}. \end{aligned}$$

Again, we integrate the frailties and obtain

$$\begin{aligned} L(\boldsymbol{\theta}|X, r = 0) \;&=\; \prod_{i=1}^{n} \int_0^{\infty} P(T = t, \delta_i = k|\omega_i, X)P(\omega_i|\theta)d\omega_i \\ &=\; \prod_{i=1}^{n} \frac{\theta^{-\sum_{k=1}^{K} I(\delta_i=k)} \theta^{-\theta^{-1}} \prod_{k=1}^{K} \left[\lambda_k(t)\exp(\beta_k^T X_i)\right]^{I(\delta_i=k)}}{\left(\sum_{k=1}^{K} \Lambda_k(t)\exp(\beta_k^T X_i) + \theta^{-1}\right)^{\theta^{-1}+\sum_{k=1}^{K} I(\delta_i=k)}}. \qquad (3.8) \end{aligned}$$

Therefore, for all valid values of $r$, our likelihood is given by combining equations (3.6) and (3.8):

$$L(\boldsymbol{\theta}|X) = \begin{cases} \prod_{i=1}^{n} \frac{\theta^{-\sum_{k=1}^{K} I(\delta_i=k)} \theta^{-\theta^{-1}} \prod_{k=1}^{K} \left[\lambda_k(t)\exp(\beta_k^T X_i)\right]^{I(\delta_i=k)}}{\left(\sum_{k=1}^{K} \Lambda_k(t)\exp(\beta_k^T X_i)+\theta^{-1}\right)^{\theta^{-1}+\sum_{k=1}^{K} I(\delta_i=k)}} & r = 0 \\ \prod_{i=1}^{n} \frac{\theta^{-1}\sum_{k=1}^{K}{}^{I(\delta_i=k)} \theta^{-\theta^{-1}} \prod_{k=1}^{K} \left[\frac{\lambda_k(t)\exp(\beta_k^T X_i)}{1+r\Lambda_k(t)\exp(\beta_k^T X_i)}\right]^{I(\delta_i=k)}}{\left(\theta^{-1}+\sum_{k=1}^{K} \log\left(1+r\Lambda_k(t)\exp(\beta_k^T X_i)\right)/r\right)^{\sum_{k=1}^{K} I(\delta_i=k)+\theta^{-1}}} & r > 0. \end{cases}$$

This is the most general form of the likelihood and is left in this expression for clarity. Extended derivation details are included in the supplemental materials. In this paper, we consider a parametric Weibull formation of the hazard, which exhibits a monotone hazard over time and is especially significant because it may be parameterized in a proportional hazards or accelerated failure time specification (Klein and Moeschberger, 2005). Mathematically, the baseline hazard takes the form $\lambda_k(t) = \gamma_k \xi_k t^{\xi_k - 1}$ and the cumulative hazard function is given by $\Lambda_k(t) = \gamma_k t^{\xi_k}$. At the end of this paper, we consider semi-parametric formulation.

### 3.3.3 Combination of effect sizes using external data

Recall that in the motivating TABS study, although the severity order for different types of events is clearly defined, the actual level of severity (i.e. the severity score) is not available in the medical claims data. Therefore, an additional analysis was conducted using data from a prospective study, the Tennessee Children's Respiratory Initiative (TCRI), to correlate the ordinal outcome with a severity score (Hartert et al., 2010). In this paper, we use this additional source of data to develop a weight function to combine the individual exposure effects in order to estimate the overall effect across all types of events. We combine the posterior distributions in a weighted fashion such that

$$\beta_{comb} = \sum_{k=1}^{K} w_k \beta_k \tag{3.9}$$

where $w_k$ is the weight for each type of effect. In particular, within each category, the sample mean is calculated and standardized over all levels. That is to say, we calculate weights as

$$\hat{w}_k = \frac{1}{n_k} \sum_{j=1}^{n_k} Score_j / \sum_{k=1}^{K} \frac{1}{n_k} \sum_{j=1}^{n_k} Score_j. \tag{3.10}$$

By replacing $w_k$ in equation (3.9) with $\hat{w}_k$, and replacing $\beta_k$ with posterior samples from the frailty model we can estimate the overall exposure effect by $\hat{\beta}_{comb}$ adjusting for disease severity. To account for the variability of $\hat{w}$ in $\hat{\beta}_{comb}$, we perform a non-parametric quantile bootstrap of the conditional standardized means (Efron, 1992).

## 3.4 Inference

In Bayesian analysis, the most common strategy for computing posterior distributions and quantities of interest is through Markov Chain Monte Carlo (MCMC) wherein an ergodic Markov Chain is constructed in such a manner that its stationary distribution is the posterior distribution of interest, $p(\boldsymbol{\theta}|X)$ (Gelman et al., 2014). Recent advances in the MCMC literature have led to the development of more efficient sampling schemes including Hamiltonian Monte Carlo and No U-Turn Sampling (Hoffman and Gelman, 2014). Certainly, the area of Bayesian survival analysis has a multitude of computational development of Gibbs sampling methods developed for specific models (Ibrahim et al., 2005). Despite these advances, however, MCMC methods remain computationally intensive for more than a few thousand observations.

Ideally, we would like to leverage the vast amount of sample information to create a richer model than we otherwise would have been able to construct using a smaller sample. In the context of linear models, model richness is gained by the provision of more spendable degrees of freedom due to increased sample information. Of course, increases in model complexity require a corresponding increase in computational resources due to the expanded dimension of the parameter space (not only with the parameter themselves, but with the correlation between those parameters). Any increases in the dimensionality of the parameter space will result in an inflation in required MCMC iterations to ensure proper mixing. Additionally, the baseline increase in wall time due to the computationally expensive evaluation of the likelihood in any given MCMC step.

### 3.4.1 Variational Bayes

The preceding facts pose a substantial challenge especially in the context of the application of survival analysis techniques to large observational studies. Fortunately, new alternatives to MCMC exist which scale Bayesian methods to large datasets using a specialized optimization framework, known as stochastic variational Bayes (Hoffman et al. (2013)).

As an alternative to sampling from an ergodic Markov Chain, variational Bayesian methods offer an inferential approach that approximates the posterior distribution through the minimization of the relative entropy between the true posterior distribution and some approximating family. For our inferential scheme, we intend to perform posterior inference over the vector $\boldsymbol{\theta} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k, \xi_1, \ldots, \xi_k, \rho_1, \ldots, \rho_k, \theta\}$. For our posterior distribution $p(\boldsymbol{\theta}|X)$, and some approximating family $q(\boldsymbol{\theta})$, the variational objective is defined by

$$q(\theta)^* = \arg\min_{q(\theta)\in\mathcal{Q}} D_{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|x)) \tag{3.11}$$

where $D_{KL}(P||Q)$ is the Kullback-Leibler divergence defined as

$$D_{KL}(P||Q) = \int \log\left(\frac{dP}{dQ}\right)\frac{dP}{dQ}dQ$$

where $P$ and $Q$ are probability measures over $\boldsymbol{\theta}$.

Our objective function therefore has an intuitive task- i.e., we seek to find the member of a particular approximating family that minimizes the statistical distance from the true posterior distribution; hence, posterior inference is converted from a sampling problem to an optimization problem. The $D_{KL}$ is a commonly used pseudo-metric of statistical distance due to its ease of interpretation. Furthermore, using $D_{KL}$ in an objective function offers a convenient way to perform the optimization. Using Jensen's inequality, it can be shown that that maximizing the Evidence Lower Bound (ELBO) minimizes $D_{KL}$; see Blei et al. (2017) for details.

It is worth mentioning the choice of approximating distribution in the variational inference routine. One of the main advantages of using a minimization criterion such as $D_{KL}$ is that no matter what distributional form is chosen, we are guaranteed that it is closest to the true posterior *in terms of the distance*. That being said, the Gaussian variational approximation is by far the most straightforward to implement. Historically, mean-field (i.e. fully-factorized Gaussian) and full-rank Gaussian distributions have been proposed, (Beal (2003), Kucukelbir et al. (2015)). Respectively, these approximations constitute taking $q(\boldsymbol{\theta}) = \prod_{\boldsymbol{\theta}} N(\mu, \sigma^2)$ and $q(\boldsymbol{\theta}) = N(\boldsymbol{\mu}, \Sigma)$ as the approximating families, respectively.

To implement variational inference, we use automatic differentiation variational inference (ADVI) (Kucukelbir et al., 2015). While in the past, variational inference has been quite difficult to implement, even for the mean-field approach, ADVI provides an automated way of specifying a model and performing the inference by first transforming the parameter space to common support and scaling. If $J$ is the number of parameters in the model, then ADVI has complexity $\mathcal{O}(NMJ)$ where $M$ is the number of Monte Carlo samples at each iteration. To perform the optimization on large datasets (potentially for datasets that cannot fit into local memory at once), one may use stochastic gradient methods in tandem with the optimization (Hoffman et al., 2013). In these methods, the gradient necessary for the optimization of the objective in equation (3.11) is approximated by its evaluation at a subset of the full data (which is called a mini-batch). By using mini-batch ADVI, the computational complexity is reduced to $\mathcal{O}(BMJ)$ where $B << N$ is the mini-batch size (Kucukelbir et al., 2015).

## 3.5 Simulations

To evaluate the validity of the proposed approach, we perform several simulation studies. In this section, we present the results of two rounds of simulations. First, we evaluate the validity of ADVI under a fixed transformation of $r$, and we follow up with some simulations regarding the choice of $r$. In all of the presented simulations, the full-rank Gaussian family is used. During initial simulation studies, we found posterior inference for $r$ to be difficult using a full-rank Gaussian. Notably, optimization of the $D_{KL}$ did not converge properly. In part, we can see from the model specification that the frailty for each person is multiplied by the transformation parameter. Because data in each cluster (i.e. each person) are limited, the correlation between these two parameters is quite difficult to estimate due to the fact that at most one event per person can be observed. Hence, the off-diagonal in the full-rank matrix is difficult to find as well.

For the simulation studies, we generated data from non-proportional hazards models of the form seen in equation (3.3). Simulation strategies for proportional hazards may be found in Bender et al. (2005) and we extend this methodology to the general transformation model. We simulated cohort covariates for that acted as proxies for age and sex. Specifically, for the age covariate, we drew $X_{age} \sim \Gamma(\alpha = 10, \beta = .3)$ and sex was generated by a simple binomial variable with success probability $p = .5$. We performed simulations for three ordinal levels of severity (i.e. $K = 3$) over fixed transformations given by $r = 0, 0.5, 1.0$ and 1.5. Parameters for estimation were given by $\boldsymbol{\beta_1} = (0.63, 0.03)^T$, $\boldsymbol{\beta_2} = (0.8, 0.03)^T$, $\boldsymbol{\beta_3} = (0.9, 0.03)^T$, $\boldsymbol{\gamma} = (0.0047, 0.0037, 0.0057)^T$, and $\boldsymbol{\xi} = (0.716, 0.725, 0.73)^T$. Vague gamma priors were given for baseline hazard and frailty parameters (Gamma(0.001, 0.001)), while diffuse Normal distributions were used for the regression coefficients (zero mean and standard deviation of 100). Note that our modeling framework allows for any specification of prior distribution and we are not restricted to a specific form.

First, for computational convenience, the log-likelihood was used for numeric stability. To perform the variational inference, we used the “variational” submodule in the package PyMC3 Bayesian statistical software written in the Python 3 programming language (Salvatier J, 2016). Python 3 is an open source computing framework and PyMC3 is a probabilistic programming package that uses the deep learning library Theano (Bastien et al., 2012) as a computational backend. By using Theano, inference in PyMC3 is aided by automatic computation of derivatives through symbolic differentiation. The procedure is accomplished by the construction of a computing graph with defined tensors for the ordinal outcomes, the covariates, and the failure times. For each of the 500 simulations, we generated a dataset of size 150,000 using the covariate and failure time generation mechanisms that were described in the previous section. Mini-batches of size 3000 were used for full-rank simulations. All simulations were ran on cluster nodes containing Intel Xeon E5-2630v3 CPUs with a base clock frequencies of 2.4GHz with 20 GB of RAM allocated to each job. For software, we used PyMC3 version 3.1 and Theano version 0.9.0.

### 3.5.1 Inference of regression coefficients with fixed transformations

The first operating characteristic of the model that we establish is the ability to find $\beta$ estimates with low bias under knowledge of the true transformation. We fix the transformation here in order to show that under proper selection of $r$, the remaining model can be well estimated. Table 3.1 contains simulation results for the regression coefficient estimates. We can see that the regression coefficient estimates across all simulation scenarios exhibit low bias (no greater than $2.93 \times 10^{-3}$) and low MSE (no greater than $2.70 \times 10^{-4}$) while maintaining coverage probability at least the nominal level.

### 3.5.2 Model Selection and selection of transformation parameter $r$

We have established that under the proper specification of transformation, estimates for the effect sizes can be well estimated using full-rank ADVI. We now focus our attention to the specification of the parameter $r$. In this section, we discuss a strategy for determining a reasonable transformation estimate using the maximum *a posteriori* (MAP) value (Rasmussen and Williams, 2006), which corresponds to the posterior mode of $r$.

| Transformation | Param. | Truth | AVPM | SDPM | SDS | MSE | CP |
|---|---|---|---|---|---|---|---|
| | | | (×100) | (×100) | (×100) | (×100) | (×100) |
| $G_k(x) = x$ | $\beta_{11}$ | 0.63 | 0.090 | 1.290 | 1.560 | 0.017 | 96.0 |
| | $\beta_{12}$ | 0.03 | -0.015 | 0.081 | 0.252 | $< 0.001$ | 100.0 |
| | $\beta_{21}$ | 0.8 | 0.065 | 1.449 | 1.630 | 0.021 | 95.4 |
| | $\beta_{22}$ | 0.03 | -0.018 | 0.084 | 0.263 | $< 0.001$ | 100.0 |
| | $\beta_{31}$ | 0.9 | 0.122 | 1.105 | 1.399 | 0.012 | 98.4 |
| | $\beta_{32}$ | 0.03 | -0.010 | 0.070 | 0.252 | $< 0.001$ | 100.0 |
| $G_k(x) = \frac{1}{0.5}\log(1 + 0.5x)$ | $\beta_{11}$ | 0.63 | 0.236 | 1.389 | 1.667 | 0.020 | 97.6 |
| | $\beta_{12}$ | 0.03 | $< 0.001$ | 0.082 | 0.259 | $< 0.001$ | 100.0 |
| | $\beta_{21}$ | 0.8 | 0.236 | 1.439 | 1.733 | 0.021 | 96.4 |
| | $\beta_{22}$ | 0.03 | -0.009 | 0.094 | 0.267 | $< 0.001$ | 100.0 |
| | $\beta_{31}$ | 0.9 | 0.221 | 1.285 | 1.576 | 0.017 | 97.8 |
| | $\beta_{32}$ | 0.03 | -0.003 | 0.083 | 0.255 | $< 0.001$ | 100.0 |
| $G_k(x) = \log(1 + x)$ | $\beta_{11}$ | 0.63 | 0.144 | 1.525 | 1.795 | 0.023 | 96.8 |
| | $\beta_{12}$ | 0.03 | 0.004 | 0.091 | 0.269 | $< 0.001$ | 100.0 |
| | $\beta_{21}$ | 0.8 | 0.086 | 1.541 | 1.856 | 0.024 | 97.8 |
| | $\beta_{22}$ | 0.03 | 0.002 | 0.097 | 0.269 | $< 0.001$ | 100.0 |
| | $\beta_{31}$ | 0.9 | 0.090 | 1.389 | 1.695 | 0.019 | 97.4 |
| | $\beta_{32}$ | 0.03 | 0.007 | 0.090 | 0.262 | $< 0.001$ | 100.0 |
| $G_k(x) = \frac{1}{1.5}\log(1 + 1.5x)$ | $\beta_{11}$ | 0.63 | 0.207 | 1.624 | 1.888 | 0.027 | 96.0 |
| | $\beta_{12}$ | 0.03 | 0.011 | 0.099 | 0.272 | $< 0.001$ | 100.0 |
| | $\beta_{21}$ | 0.8 | 0.213 | 1.717 | 1.942 | 0.030 | 95.2 |
| | $\beta_{22}$ | 0.03 | 0.013 | 0.108 | 0.281 | $< 0.001$ | 100.0 |
| | $\beta_{31}$ | 0.9 | 0.293 | 1.472 | 1.827 | 0.023 | 97.4 |
| | $\beta_{32}$ | 0.03 | 0.006 | 0.092 | 0.268 | $< 0.001$ | 100.0 |

Table 3.1: Full Rank ADVI simulation results with $r$ fixed at the truth

Instead of modeling $r$ as part of a whole approximate Bayesian solution, we propose to fix the value at the MAP value. We perform simulations under this scenario and explore how well variational inference performs for the rest of the model. Secondly, we define a neighborhood around the MAP estimate and perform model selection based on information criteria over the neighborhood grid of values. Additionally, in the context of scientific inference, interpretability of coefficients is an important aspect for researchers. In the context of this transformation model, we have two special cases that can be easily interpreted by researchers. Specifically, when $r = 0$, we recover the well known parametric proportional hazards model and when $r = 1$, we recover the proportional odds model. As such, in practice, a researcher could fit the model developed in this framework with $\hat{r}$ along with the two interpretable transformations using information criteria. Here, we use the deviance information criteria (DIC), Watanabe-Akaike Information Criteria (WAIC), and the Bayesian Predictive

Information Criteria (BPIC).

First, the DIC (Spiegelhalter et al., 2002) is a widely used measure of model fit defined as

$$DIC = D(\bar{\boldsymbol{\theta}}) + 2p_D$$

where $D(\boldsymbol{\theta}) = -2\log L(\boldsymbol{\theta}|X) + C$, and $p_D = \bar{D} - D(\bar{\boldsymbol{\theta}})$. Secondly, the WAIC can be thought of as a large sample approximation to cross-validation, and is defined as

$$WAIC = -2\left(\sum_{i=1}^{n}\log\left(\frac{1}{S}\sum_{s=1}^{S}p(y_i|\theta^s)\right) - \sum_{i=1}^{n}V_{s=1}^{S}(\log(p(y_i|\theta^s)))\right)$$

where $V_{s=1}^{S}a_s = \frac{1}{S-1}\sum_{s=1}^{S}(a_s - \bar{s})^2$ and is generally more stable because it computes the variance separately for each data point and then performs the summation (Vehtari et al., 2017). The final information criteria that we consider is the Bayesian Predictive information criteria (BPIC) which was developed as an estimator of the posterior mean of the expected log-likelihood of the predictive distribution when the specified family of probability distributions does not contain the true distribution (Ando, 2007).

In Table 3.2 we can see results of model estimation when we fix the transformation parameter to the optimized MAP value. We notice that the bias of the posterior means is very similar across all levels of true transformation parameter when compared to the results in table 3.1. Any increase in the average bias is small, however, with no more than a $1.38 \times 10^{-3}$ difference in parameter bias from the previous simulations performed under knowledge of the true $r$. Additionally, coverage probability has been reduced for some of the regression coefficients with 93.8% coverage being the lowest among all of the posterior distributions.

### 3.5.3 Model Selection in an $\epsilon$-neighborhood of the MAP estimate

We consider the selection of a model in a neighborhood surrounding the MAP estimate. That is to say, we first compute the MAP estimate, $\hat{r}$, and subsequently

| Transformation | Param. | Truth | AVPM | SDPM | SDS | MSE | CP |
|---|---|---|---|---|---|---|---|
| | | | (×100) | (×100) | (×100) | (×100) | (×100) |
| $G_k(x) = x$ | $\beta_{11}$ | 0.63 | 0.061 | 1.284 | 1.533 | 0.017 | 96.8 |
| | $\beta_{12}$ | 0.03 | -0.012 | 0.084 | 0.245 | $< 0.001$ | 100.0 |
| | $\beta_{21}$ | 0.8 | -0.126 | 1.465 | 1.603 | 0.022 | 95.4 |
| | $\beta_{22}$ | 0.03 | -0.02 | 0.092 | 0.262 | $< 0.001$ | 100.0 |
| | $\beta_{31}$ | 0.9 | 0.213 | 1.08 | 1.407 | 0.012 | 98.0 |
| | $\beta_{32}$ | 0.03 | -0.012 | 0.081 | 0.244 | $< 0.001$ | 100.0 |
| $G_k(x) = \frac{1}{0.5}\log(1 + 0.5x)$ | $\beta_{11}$ | 0.63 | 0.136 | 1.574 | 1.658 | 0.025 | 95.6 |
| | $\beta_{12}$ | 0.03 | -0.009 | 0.089 | 0.258 | $< 0.001$ | 100.0 |
| | $\beta_{21}$ | 0.8 | 0.278 | 1.534 | 1.745 | 0.024 | 96.2 |
| | $\beta_{22}$ | 0.03 | -0.002 | 0.094 | 0.267 | $< 0.001$ | 100.0 |
| | $\beta_{31}$ | 0.9 | 0.257 | 1.364 | 1.564 | 0.019 | 97.0 |
| | $\beta_{32}$ | 0.03 | -0.0 | 0.083 | 0.252 | $< 0.001$ | 100.0 |
| $G_k(x) = \log(1 + x)$ | $\beta_{11}$ | 0.63 | 0.196 | 1.586 | 1.778 | 0.026 | 96.6 |
| | $\beta_{12}$ | 0.03 | 0.008 | 0.096 | 0.263 | $< 0.001$ | 100.0 |
| | $\beta_{21}$ | 0.8 | 0.224 | 1.670 | 1.880 | 0.028 | 96.2 |
| | $\beta_{22}$ | 0.03 | -0.004 | 0.093 | 0.274 | $< 0.001$ | 100.0 |
| | $\beta_{31}$ | 0.9 | 0.159 | 1.646 | 1.691 | 0.027 | 94.2 |
| | $\beta_{32}$ | 0.03 | 0.002 | 0.090 | 0.259 | $< 0.001$ | 100.0 |
| $G_k(x) = \frac{1}{1.5}\log(1 + 1.5x)$ | $\beta_{11}$ | 0.63 | 0.247 | 1.685 | 1.863 | 0.029 | 96.2 |
| | $\beta_{12}$ | 0.03 | 0.009 | 0.093 | 0.266 | $< 0.001$ | 99.8 |
| | $\beta_{21}$ | 0.8 | 0.295 | 1.608 | 1.956 | 0.027 | 98.4 |
| | $\beta_{22}$ | 0.03 | 0.009 | 0.099 | 0.276 | $< 0.001$ | 100.0 |
| | $\beta_{31}$ | 0.9 | 0.275 | 1.906 | 1.817 | 0.037 | 93.8 |
| | $\beta_{32}$ | 0.03 | 0.004 | 0.092 | 0.262 | $< 0.001$ | 100.0 |

Table 3.2: Full Rank ADVI simulation results fixing $r$ at the MAP estimate

search the space of models over an evenly spaced grid for the neighborhood $[\hat{r}-\epsilon, \hat{r}+\epsilon]$, evaluating each point with the information criteria outlined in the preceding section. We perform the grid search over $\epsilon-$ neighborhoods of size 0.14. Figure 3.2 presents kernel density plots of the transformation parameters selected by this approach. We can see that this selection criteria provides a reasonable selection method for $r$. In each simulation, the mode of the selected values correspond to the underlying true simulated transformation parameter.

### 3.5.4 Inference for frailty parameter

During the course of the simulations, we noticed that inference for the frailty parameter exhibited interesting behavior. Specifically, we found that the posterior means of the gamma frailty parameter, $\theta$ had low bias but exhibited under-coverage in the credible interval. We notice that the standard deviation of the posterior mean is

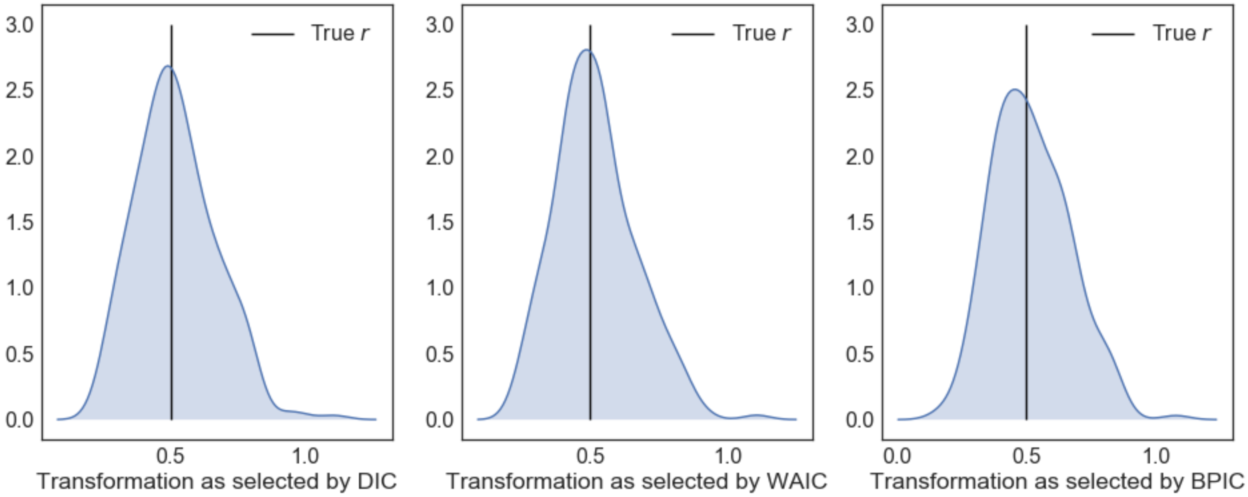


Figure 3.2: Selection of transformation parameter in $\epsilon-$ neighborhood of $\hat{r}$,

| True Model | ABPM ($\times$100) | SDPM ($\times$100) | SDS ($\times$100) | MSE ($\times$100) | CP ($\times$100) |
|---|---|---|---|---|---|
| TTFE1 | 0.702 | 1.232 | 1.327 | 0.02 | 94.0 |
| TTFE2 | 0.837 | 6.704 | 1.305 | 0.456 | 28.0 |
| TTME | 0.338 | 0.58 | 0.669 | 0.005 | 94.2 |

Table 3.3: Inference for frailty parameter, $\theta$, comparing time to first event with $r$ fixed at the truth (TTFE1), $r$ estimated at the MAP estimate (TTFE2), and time to multiple events (TTME). Simulated under $r = 0.5$.

smaller than the standard deviation of the simulations; indicating that the variation in the point estimate for $\theta$ conditional on the optimized $r$ was greater than the variation found in each posterior distribution on average. We hypothesized that this extra variation was due the fact that time-to-first event resulted in little between-person information. To test this, we ran a simulation that allowed for any number of events to be observed within a person (see Appendix for derivation details). Table 3.3 shows the results of these simulations for the case with the most extreme under-coverage of the posterior credible interval. We can see that the simulation confirmed our hypothesis; the coverage probability is closer to the nominal level and the standard deviation of the posterior mean is smaller than the time-to-first event case.

In essence, we can interpret these results in the context of 'no free lunch.' Specifically, due to the decreased within person information, there is increased variability in the frailty parameter estimates. Notably, since the frailty terms for each cluster are multiplied by the transformation parameter, the increase in variability in the frailty is correlated with the transformation parameter. Now, since $r$ and $\theta$ are multiplied mathematically, the fact that we fix one parameter at a point estimate results in the under estimation of the variability of posterior estimates of the frailty parameter. We notice that the difference in the average bias between posterior mean the model that is fixed at the true transformation parameter and the model that fixes $r$ at the MAP estimate is very small ($1.35 \times 10^{-3}$). The variability that is neglected by fixing the transformation parameter at the MAP estimate is transferred to the variability in the point estimate (the posterior mean) of $\theta$. The under-coverage of the posterior credible interval is attributable to the under-estimation of the variability in $\theta$ when fixing $r$.

## 3.6 Application

Bronchiolitis is an infection of the lower respiratory tract most commonly occurring in infants and children under two years of age. Respiratory syncytial virus (RSV) is the most common cause of infant bronchiolitis with peak ages of two to six months; up to 50-80% of bronchiolitis episodes during infancy are due to RSV infection (Meissner, 2016). Bronchiolitis is an important medical condition to study not only in its own right, but due to the fact that it has itself been identified as an important risk factor for subsequent pediatric respiratory conditions (Wu et al., 2008). As a motivating problem for this paper, it is of interest to study the association between maternal smoking during pregnancy and infant bronchiolitis severity.

As an application of our model, a de-identified dataset from an established birth cohort of infants and their biological mothers enrolled in TennCare and designed to assess the relationship between infant bronchiolitis and childhood asthma, Tennessee Asthma Bronchiolitis Study cohort (TABS) was utilized. The de-identified dataset included 163,763 patient records was obtained and analyzed. In this dataset, three levels of severity were included for bronchiolitis visit: out-patient visit (“opv”), emergency room visit (“emr”) and hospitalization (“hos”) (Carroll et al., 2009). The percentages of patients who experienced each of the aforementioned ordinal levels were 10.7% 6.2%, and 4.0%, respectively. About 79.1% of patients in the cohort did not experience an event within the study timeframe (first year of life). Covariates that were included in the model include delivery method, number of living siblings related to the infant, indicators for residence, infant race, and standardized birthweight in grams.

Following the analysis of the TABS cohort data, follow-up validation data validation was assessed in the TCRI cohort, a prospective cohort in which infants with acute respiratory illness were enrolled, and disease severity quantified using a respiratory severity score. In this cohort, the Tal score (McCallum et al., 2013), which is a simple respiratory severity score that has been demonstrated to have good inter-rater reliability, was calculated, and level of healthcare utilization, hospitalization, emergency department visit, or unscheduled outpatient visit recorded. The respiratory severity score ranges from 0 to 12, with a higher score indicating more severe disease. This score is derived as an aggregate of assigned values ranging from 0 to 3 in categories of respiratory rate, retractions, wheezing, and oxygen saturation in room air and has been shown to discriminate level of healthcare utilization as well as lower versus upper respiratory tract infection as another marker of disease severity (Rodriguez et al., 2016). It should be noted that the individuals contained in this follow-up study are not the same subjects contained in the TennCare dataset. The study was approved by the Institutional Review Board of the Vanderbilt University and the Tennessee Department of Health.

| | Est. | HR | SE | z | p | Lower 95% CI | Upper 95% CI |
|---|---|---|---|---|---|---|---|
| genderMale | 0.290 | 1.336 | 0.011 | 26.263 | 0.000 | 0.268 | 0.311 |
| masthma | 0.134 | 1.143 | 0.025 | 5.419 | 0.000 | 0.085 | 0.182 |
| mat-smoke | 0.131 | 1.140 | 0.012 | 10.719 | 0.000 | 0.107 | 0.155 |
| assiss | -0.008 | 0.992 | 0.021 | -0.363 | 0.716 | -0.050 | 0.034 |
| c-section | 0.051 | 1.052 | 0.013 | 3.957 | 0.000 | 0.026 | 0.076 |
| AfAm | -0.241 | 0.786 | 0.015 | -16.111 | 0.000 | -0.270 | -0.211 |
| residence2 | -0.051 | 0.950 | 0.014 | -3.678 | 0.000 | -0.078 | -0.024 |
| residence3 | -0.204 | 0.815 | 0.015 | -13.389 | 0.000 | -0.234 | -0.174 |
| weight$_{s}tand$ | -0.068 | 0.934 | 0.006 | -12.015 | 0.000 | -0.079 | -0.057 |
| siblings | 0.031 | 1.031 | 0.004 | 7.246 | 0.000 | 0.023 | 0.039 |

Table 3.4: Univariate Cox proportional hazards model

### 3.6.1 Univariate Analyses

A preliminary analysis was performed assessing the time to any first event as a univariate cox proportional hazards model. Table 3.4 contains the fits of the univariate Cox model.

We see that according to this model, the hazard ratio for time to first bronchiolitis episode comparing mothers who smoked during pregnancy to those that didn't is Mean: 1.140, 95% CI: (1.110, 1.168). We will compare this to the severity adjusted quantity using the secondary dataset.

### 3.6.2 Computing Time and Implementation

The proposed method was used to analyze the data collected in the TABS study. Our first attempt at model fitting involved the usage of No U-Turn sampling using PyMC3. Using a computing cluster, we allocated 8 Xeon E5-2630v3 CPU cores and 150 GB of RAM to running MCMC. As such, we ran MCMC on the dataset 10 times to benchmark the performance. We ran each instance for 24 hours and measured sampling rate as well as the projected time to draw 10000 samples. After 24 hours, MCMC sampled at an average of 30.89 seconds per draw. The average projected time to completion was 91.10 hours (3.8 days).

Next, we fit the model using the proposed ADVI framework. For ADVI, usage of the cluster was not necessary for model fitting; we were able to successfully run 40,000 ADVI iterations in 27.62 minutes on a standard laptop equipped with a 2.4 GHz Intel Core i5 processor and 8 GB of memory. Using this same local machine hardware, however, employing MCMC invariably resulted in a system hang. In the final analysis, however, we conservatively ran ADVI for a total of 150,000 iterations to ensure convergence which took an average of 125.9 minutes to complete.

### 3.6.3 Results

The transformation parameter was found to be optimized at $\hat{r} = 0.24$. Bootstrapping the validated Tal scores from the TCRI cohort resulted in symmetric weight distributions (we used 1000 bootstrap replications). Table 3.5 presents the results of fitting the model with model selection criteria.

With respect to the interpretable quantities, we can see that the proportional hazards model ($r = 0$) is slightly preferred over the proportional hazards odds ($r = 1$), with the exception of WAIC (although this difference is quite small). These quantities suggest, however, that the transformation has little effect on the inference of the regression coefficients with respect to overall model fit. Since the proportional hazards model was the preferred interpretable model preferred by both DIC and BPIC, we present model fit results from this transformation along with those fitted with $\hat{r}$ in Table 3.6. We see very similar results in the effect sizes as that presented with optimized $r$. The individual smoking effects in outpatient, emergency room, and hospitalization were combined using equation (3.9) using the bootstrapped $\hat{w}_k$s. Specifically, when combined with severity scores according to equation (3.9), we interpret the exponentiated combined score (Mean: 1.243, 95% Credible Interval (1.160 1.333)) as the severity-weighted hazard ratio of infant bronchiolitis comparing mothers who smoked during pregnancy versus those did not smoke, adjusting for other covariates.

In essence, this quantity provides a score that is informative for health policy makers: it describes the weighting scheme provides an indication of disease burden. Figure 3.3 compares the distributions of the exponentiated combined regression coefficients for $r = 0$ and $r = \hat{r}$; we notice that the two distributions are quite similar. If we compare this to the full MCMC implementation, we have Mean: 1.271, 95% CI (1.189,1.359). Further, we observe only a slight overlap in the interval estimates comparing the combined score to the univariate Cox PH model discussed earlier.

| Model | WAIC | DIC | BPIC |
|---|---|---|---|
| Proportional Hazards, $r = 0$ | 567688.61 | 632938.35 | 616691.63 |
| Proportional Odds, $r = 1$ | 567307.66 | 637702.07 | 624291.48 |
| $\hat{r}$ | 567315.32 | 633034.14 | 616939.19 |

Table 3.5: Information criteria comparing fitted models with $r = 0$, and $r = 1$, and $r = \hat{r}$

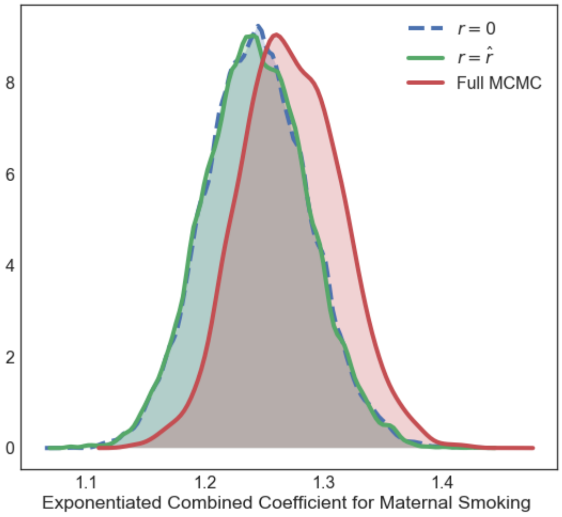


Figure 3.3: Combined exponentiated regression coefficient comparing $r = 0$ and $r = \hat{r}$

## 3.7 Discussion

We have presented a model that systematically and flexibly models time to first event for ordinal disease severity. Extending the method to multivariate ordinal

|  |  | $r = 0$ |  |  |  | $r = \hat{r}$ |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| Type of Outcome |  | Post. Mean | Post. SD | 2.5th HPD | 97.5th HPD | Post. Mean | Post. SD | 2.5th Quant. | 97.5th Quant |
| Emergency Room | $\beta_{00}$ | 0.550 | 0.037 | 0.477 | 0.621 | 0.549 | 0.040 | 0.473 | 0.627 |
|  | $\beta_{01}$ | 0.314 | 0.071 | 0.175 | 0.453 | 0.310 | 0.076 | 0.163 | 0.460 |
|  | $\beta_{02}$ | 0.268 | 0.040 | 0.190 | 0.347 | 0.272 | 0.040 | 0.193 | 0.351 |
|  | $\beta_{03}$ | 0.056 | 0.015 | 0.027 | 0.086 | 0.062 | 0.018 | 0.027 | 0.098 |
|  | $\beta_{04}$ | -0.097 | 0.063 | -0.221 | 0.027 | -0.093 | 0.066 | -0.222 | 0.036 |
|  | $\beta_{05}$ | 0.058 | 0.041 | -0.023 | 0.139 | 0.063 | 0.033 | -0.002 | 0.129 |
|  | $\beta_{06}$ | -0.263 | 0.043 | -0.345 | -0.180 | -0.248 | 0.040 | -0.326 | -0.170 |
|  | $\beta_{07}$ | 0.029 | 0.044 | -0.058 | 0.115 | 0.048 | 0.044 | -0.039 | 0.133 |
|  | $\beta_{08}$ | 0.199 | 0.043 | 0.116 | 0.282 | 0.230 | 0.048 | 0.136 | 0.323 |
|  | $\beta_{09}$ | -0.130 | 0.020 | -0.169 | -0.091 | -0.122 | 0.018 | -0.156 | -0.087 |
| Hospitalization | $\beta_{10}$ | 0.436 | 0.034 | 0.369 | 0.502 | 0.427 | 0.038 | 0.352 | 0.502 |
|  | $\beta_{11}$ | 0.146 | 0.087 | -0.025 | 0.315 | 0.140 | 0.085 | -0.025 | 0.306 |
|  | $\beta_{12}$ | 0.226 | 0.042 | 0.143 | 0.309 | 0.225 | 0.045 | 0.136 | 0.311 |
|  | $\beta_{13}$ | 0.151 | 0.021 | 0.109 | 0.191 | 0.153 | 0.016 | 0.121 | 0.184 |
|  | $\beta_{14}$ | -0.051 | 0.068 | -0.185 | 0.084 | -0.049 | 0.081 | -0.207 | 0.111 |
|  | $\beta_{15}$ | 0.092 | 0.043 | 0.009 | 0.175 | 0.096 | 0.040 | 0.018 | 0.174 |
|  | $\beta_{16}$ | -0.598 | 0.044 | -0.685 | -0.515 | -0.587 | 0.048 | -0.681 | -0.493 |
|  | $\beta_{17}$ | -0.416 | 0.047 | -0.507 | -0.324 | -0.409 | 0.043 | -0.494 | -0.323 |
|  | $\beta_{18}$ | -0.707 | 0.044 | -0.794 | -0.621 | -0.696 | 0.050 | -0.793 | -0.598 |
|  | $\beta_{19}$ | -0.133 | 0.020 | -0.173 | -0.093 | -0.125 | 0.020 | -0.164 | -0.087 |
| Outpatient Visit | $\beta_{20}$ | 0.530 | 0.030 | 0.472 | 0.589 | 0.531 | 0.035 | 0.462 | 0.598 |
|  | $\beta_{21}$ | 0.098 | 0.066 | -0.031 | 0.228 | 0.088 | 0.071 | -0.049 | 0.228 |
|  | $\beta_{22}$ | 0.077 | 0.040 | -0.002 | 0.154 | 0.069 | 0.038 | -0.007 | 0.144 |
|  | $\beta_{23}$ | 0.059 | 0.017 | 0.026 | 0.092 | 0.058 | 0.017 | 0.026 | 0.091 |
|  | $\beta_{24}$ | -0.027 | 0.049 | -0.123 | 0.068 | -0.028 | 0.060 | -0.145 | 0.093 |
|  | $\beta_{25}$ | 0.085 | 0.035 | 0.017 | 0.156 | 0.087 | 0.040 | 0.009 | 0.166 |
|  | $\beta_{26}$ | -0.703 | 0.039 | -0.779 | -0.626 | -0.713 | 0.043 | -0.797 | -0.629 |
|  | $\beta_{27}$ | -0.018 | 0.038 | -0.092 | 0.058 | -0.005 | 0.042 | -0.088 | 0.078 |
|  | $\beta_{28}$ | -0.617 | 0.037 | -0.690 | -0.545 | -0.623 | 0.047 | -0.715 | -0.531 |
|  | $\beta_{29}$ | -0.089 | 0.018 | -0.125 | -0.053 | -0.087 | 0.019 | -0.124 | -0.051 |

Table 3.6: Posterior Quantities for Bronchiolitis cohort

survival outcomes is straightforward. The composite quantity developed in equation (3.9) offers a composite score weighted by disease severity instead of by prevalence of each disease level in the underlying study cohort. When severity score is not available, the composite score can be constructed assuming ordinal scores such as $1, 2, \ldots, K$ - a common practice in dealing with ordinality of outcomes. Note that inclusion of recurrent events is also straightforward with appropriate modification to the likelihood function. In our simulations, several insights were gained both from the perspective of the underlying model assumed for the conditional hazard, but also regarding the interface of variational approximation to the posterior distribution. Considering the operating characteristics of our methodology, we see that the model performs well under the simulated scenarios, and provides relatively small bias for most model parameters. Specifically, the effect sizes were well estimated using ADVI, which is encouraging for clinical relevance. With respect to the combination step of the effect sizes across levels of severity, we found that the bootstrap distribution of the standardized sample means were symmetric around their mean values. Certainly other quantities could have been taken from the skewed distribution and bootstrapped in a similar fashion. Additionally, in the outpatient group of the validation study, we observed what appeared to be a mixture distribution. In this case, it may be interesting to incorporate the weights via a Dirichlet process mixture.

To our knowledge, there has been no direct application of variational inference to multivariate survival models, especially in the presence of a full-rank approximation. As such, this work provides a foundation for future research in more complex survival models using variational Bayesian methods. Alternatives to the approximation paradigm of variational inference, however, do exist. Divide and conquer strategies including expectation propagation (Minka, 2001) and combination strategies using optimal transport theory (Srivastava et al., 2015) are both appealing alternatives to variational Bayes and warrant further investigation into their application to multivariate survival models.

Certainly, more rich approximations may be made to the posterior (see future directions in the conclusion of this thesis). We strongly consider the approximation made in this present work to be more statistically sound than other approximations due to the rigorous definition of the objective function for the optimization procedure, however. Specifically, Laplace's approximation to the posterior distribution is one commonly used methodology. To review, Laplace's approximation posits a normal approximation to a posterior distribution that uses information about the posterior mode and curvature to develop the approximation. Roughly speaking, the mean of the approximation is set to the maximum a posteriori estimate of $\theta$ and the variance of the approximation is calibrated by the expected information of the posterior distribution. However, it has been noted by several authors that this approximation is only correct under very stringent conditions (Geisser et al., 1990), one such condition being that the error of the approximation must be of order $O(n^{-2})$, a fact that must be known a priori.

Perhaps the most direct extension of the model described in this paper is the specification of a more flexible form of the baseline hazard function $\lambda_0(t)$. In de Castro et al. (2014), the authors use a gamma frailty model with a piecewise exponential model, specifying a piecewise baseline hazard. Alternatively, we can extend this multivariate Bayesian framework to include splines. Such models, however, employ more parameters for their non-parametric components. As such, MCMC will become even more computationally expensive, further motivating the exploration of the validity of ADVI in large-scale multivariate survival models. See the conclusion of this dissertation for a full discussion of this future direction.

While the full-rank Gaussian alternative provides the advantage that correlations between parameters are captured, the main drawback with this modification is that the approximation is restrictive in that it assumes the posterior distribution to be elliptically shaped. In practice, we would like to be able to model the posterior with an arbitrary shape for the approximating distribution while maintaining scalability with respect to the number of parameters in the model. One promising approach to overcome these challenges is the modification of the mean-field ADVI with normalizing flows (Rezende and Mohamed, 2015). Within the Gaussian approximation however, computational load can be reduced by specifying a grouped approximation where a block of the covariance of the approximating distribution is modeled as full-rank and the remainder fully-factorized. These extensions are currently under investigation and the future for adapting Bayesian methods to large scale data sets is promising.

### 3.7.1 Acknowledgement

The authors thank the Tennessee Division of TennCare of the Department of Finance and Administration, as well as the Tennessee Department of Health for providing data for the study.

## 3.8 Appendix

### 3.8.1 Derivation Details

Now, let $Y_k$ denote the true event time of the $k^{th}$ event $k = 1, \dots, K$ and $Y_0 = C$. We observe $T = \min\{Y_0, Y_1, \dots, Y_K\}$ and $\delta = \{k : Y_k \leq Y_j, j \neq k\}$. To account for correlation within an individual, we specify a frailty term, denoted by $\omega_i$. Where $\omega_i \sim \Gamma(\theta^{-1}, \theta^{-1})$. From here, we model the cumulative hazard as

$$\begin{aligned}\Lambda_{ik}(t|\omega_i, \mathbf{X}_i) &= \omega_i G_k \left\{\Lambda_k(t)\exp(\beta_k^T X_i)\right\} \\ &= \omega_i \log\left\{1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right\}/r\end{aligned}$$

From this specification one can conclude that the survival function and hazard functions are each given by

$$\begin{aligned}S_{ik}(t|\omega_i, X_i) &= \exp\left\{-\Lambda_{ik}(t|\omega_i, X_i)\right\} \\ &= \exp\left\{-\omega_i \log\left[1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right]/r\right\}\end{aligned}$$

and

$$\begin{aligned}\lambda_{ik}(t|\omega_i, X_i) &= \frac{\partial\Lambda_{ik}(t|\omega_i, X_i)}{\partial t} \\ &= \lambda_k(t)\exp(\beta_k^T X_i)\omega_i\left\{1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right\}^{-1}\end{aligned}$$

We therefore specify the joint probability distribution of the event time and $\delta$ by

$$\begin{aligned}
P(T_i = t, \delta_i = k | \omega_i, X_i, r > 0) &= \prod_{k=1}^{K} \left[ f_k(t) \prod_{j \neq k} S_j(t) \right]^{I(\delta=k)} [S_k(t)]^{I(\delta=0)} \\
&= \prod_{k=1}^{K} f_k(t)^{I(\delta=k)} S_k(t)^{1-I(\delta=k)} \\
&= \prod_{k=1}^{K} \lambda_k(t)^{I(\delta=k)} S_k(t) \\
&= \prod_{i=1}^{n} \left[ \lambda_k(t) \exp(\beta_k^T X_i) \omega_i \left\{ 1 + r\Lambda_k(t) \exp(\beta_k^T X_i) \right\}^{-1} \right]^{I(\delta=k)} \\
&\qquad \times \exp\left\{ -\omega_i \log\left\{ 1 + r\Lambda_k(t) \exp(\beta_k^T X_i)/r_k \right\} \right\}
\end{aligned}$$

We note that the frailties are nuisance parameters, so we integrate them with respect to the density of $\omega_i$ such that

$$\begin{aligned}
L(\boldsymbol{\theta}|X_i, r>0) &= \int_0^\infty P(T_i = t, \delta_i = k|\omega_i, X_i, r>0)P(\omega_i|\theta)d\omega_i \\
&= \int_0^\infty \prod_{k=1}^{K}\left[\lambda_k(t)\exp(\beta_k^T X_i)\omega_i\left\{1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right\}^{-1}\right]^{I(\delta=k)} \\
&\quad \frac{\theta^{-\theta^{-1}}}{\Gamma(\theta^{-1})}\omega_i^{\theta^{-1}-1}e^{-\theta^{-1}\omega_i} \\
&\quad \times\exp\left\{-\omega_i\sum_{k=1}^{K}\log\left(1 + r\Lambda_k(t)e^{\boldsymbol{x}_i'\boldsymbol{\beta}}\right)/r_k\right\}d\omega_i \\
&= \int_0^\infty \omega_i^{\sum_{k=1}^K I(\delta_i=k)}\omega_i^{\theta^{-1}-1}\exp\left\{-\omega_i\theta^{-1}\right\}\frac{\theta^{-\theta^{-1}}}{\Gamma(\theta^{-1})} \\
&\quad \times\exp\left\{-\omega_i\sum_{k=1}^{K}\log\left[1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right]/r_k\right\}d\omega_i \\
&\quad \times\prod_{k=1}^{K}\left[\lambda_k(t)\exp(\beta_k^T X_i)\left\{1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right\}^{-1}\right]^{I(\delta_i=k)} \\
&= \int_0^\infty \omega_i^{\sum_{k=1}^K I(\delta_i=k)+\theta^{-1}-1}\frac{\theta^{-\theta^{-1}}}{\Gamma(\theta^{-1})} \\
&\quad \times\exp\left\{-\omega_i(\theta^{-1} + \sum_{k=1}^{K}\log\left[1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right]/r)\right\}d\omega_i \\
&\quad \times\prod_{k=1}^{K}\left[\lambda_k(t)\exp(\beta_k^T X_i)\left\{1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right\}^{-1}\right]^{I(\delta_i=k)}
\end{aligned}$$

We recognize the integral in the final line as the kernel of a gamma density.

$$\begin{aligned}
L(\boldsymbol{\theta}|X_i, r>0) &= \frac{\Gamma\left(\sum_{k=1}^K I(\delta_i = k) + \theta^{-1}\right)\theta^{-\theta^{-1}}\Gamma(\theta^{-1})^{-1}}{\left(\theta^{-1} + \sum_{k=1}^K \log\left(1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right)/r_k\right)^{\sum_{k=1}^K I(\delta_i=k)+\theta^{-1}}} \\
&\quad \times\prod_{k=1}^{K}\left[\frac{\lambda_k(t)\exp(\beta_k^T X_i)}{1 + r\Lambda_k(t)\exp(\beta_k^T X_i)}\right]^{I(\delta_i=k)}
\end{aligned}$$

Furthermore, that because $\frac{\Gamma(\alpha+1)}{\Gamma(\alpha)} = \alpha$ and since $\sum_{k=1}^K I(\delta_i = k) \in \{0, 1\}$, we conclude that

$$
\begin{aligned}
L(\boldsymbol{\theta}|X_i, r > 0) &= \frac{\theta^{-\sum_{k=1}^{K} I(\delta_i = k)} \theta^{-\theta^{-1}}}{\left(\theta^{-1} + \sum_{k=1}^{K} \log\left(1 + r\Lambda_k(t)\exp(\beta_k^T X_i)\right)/r_k\right)^{\sum_{k=1}^{K} I(\delta_i=k)+\theta^{-1}}} \\
&\quad \times \prod_{k=1}^{K} \left[\frac{\lambda_k(t)\exp(\beta_k^T X_i)}{1 + r\Lambda_k(t)\exp(\beta_k^T X_i)}\right]^{I(\delta_i=k)}
\end{aligned}
$$

Now for the case where $r = 0$ we have the joint probability of

$$
\begin{aligned}
P\left(T = t, \delta = k|\omega_i, X_i, r = 0\right) &= \prod_{k=1}^{K} \lambda_k(t)^{I(\delta=k)} S_k(t) \\
&= \prod_{k=1}^{K} \left[\omega_i \lambda_k(t)\exp(\beta_k^T X_i)\right]^{I(\delta=k)} \\
&\quad \times \exp\left\{-\omega_i \Lambda_k(t)\exp(\beta_k^T X_i)\right\}
\end{aligned}
$$

Again, we integrate the frailties and obtain

$$\begin{aligned}
L(\boldsymbol{\theta}|X_i, r=0) &= \int_0^\infty P(T=t, \delta_i = k|\omega_i, X_i, r=0)P(\omega_i|\theta)d\omega_i \\
&= \int_0^\infty \prod_{k=1}^K \Big[\omega_i \lambda_k(t)\exp(\beta_k^T X_i)\Big]^{I(\delta=k)} \exp\Big\{-\omega_i \Lambda_k(t)\exp(\beta_k^T X_i)\Big\} \\
&\qquad\qquad\qquad\qquad \times \frac{\theta^{-\theta^{-1}}}{\Gamma(\theta^{-1})}\omega_i^{\theta^{-1}-1}e^{-\theta^{-1}\omega_i}d\omega_i \\
&= \int_0^\infty \omega_i^{\sum_{k=1}^K I(\delta_i=k)} e^{-\omega_i \sum_{k=1}^K \Lambda_k(t)\exp(\beta_k^T X_i)} \frac{\theta^{-\theta^{-1}}}{\Gamma(\theta^{-1})}\omega_i^{\theta^{-1}-1}e^{-\theta^{-1}\omega_i}d\omega_i \\
&\quad \times \prod_{k=1}^K \Big[\lambda_k(t)\exp(\beta_k^T X_i)\Big]^{I(\delta_i=k)} \\
&= \int_0^\infty \omega_i^{\sum_{k=1}^K I(\delta_i=k)} e^{-\omega_i \sum_{k=1}^K \Lambda_k(t)\exp(\beta_k^T X_i)} \frac{\theta^{-\theta}}{\Gamma(\theta^{-1})}\omega_i^{\theta^{-1}-1}e^{-\theta^{-1}\omega_i}d\omega_i \\
&\quad \times \prod_{k=1}^K \Big[\lambda_k(t)\exp(\beta_k^T X_i)\Big]^{I(\delta_i=k)} \\
&= \int_0^\infty \omega_i^{\sum_{k=1}^K I(\delta_i=k)+\theta^{-1}-1} e^{-\omega_i\left(\sum_{k=1}^K \Lambda_k(t)\exp(\beta_k^T X_i)+\theta^{-1}\right)}d\omega_i \frac{\theta^{-\theta^{-1}}}{\Gamma(\theta^{-1})} \\
&\quad \times \prod_{k=1}^K \Big[\lambda_k(t)\exp(\beta_k^T X_i)\Big]^{I(\delta_i=k)} \\
&= \frac{\Gamma\left(\theta^{-1}+\sum_{k=1}^K I(\delta_i=k)\right)\prod_{k=1}^K \Big[\lambda_k(t)\exp(\beta_k^T X_i)\Big]^{I(\delta_i=k)}\theta^{-\theta^{-1}}}{\left(\sum_{k=1}^K \Lambda_k(t)\exp(\beta_k^T X_i)+\theta^{-1}\right)^{\theta^{-1}+\sum_{k=1}^K I(\delta_i=k)}\Gamma(\theta^{-1})}
\end{aligned}$$

Therefore, for all valid values of $r$, our likelihood is given by

$$L(\boldsymbol{\theta}|X_i) = \begin{cases} \dfrac{\Gamma\left(\theta^{-1}+\sum_{k=1}^K I(\delta_i=k)\right)\prod_{k=1}^K\left[\lambda_k(t)\exp(\beta_k^T X_i)\right]^{I(\delta_i=k)}\theta^{-\theta^{-1}}}{\left(\sum_{k=1}^K \Lambda_k(t)\exp(\beta_k^T X_i)+\theta^{-1}\right)^{\theta^{-1}+\sum_{k=1}^K I(\delta_i=k)}\Gamma(\theta^{-1})} & r=0 \\ \dfrac{\theta^{-\sum_{k=1}^K I(\delta_i=k)}\theta^{-\theta^{-1}}\prod_{k=1}^K\left[\frac{\lambda_k(t)\exp(\beta_k^T X_i)}{1+r\Lambda_k(t)\exp(\beta_k^T X_i)}\right]^{I(\delta_i=k)}}{\left(\theta^{-1}+\sum_{k=1}^K \log\left(1+r\Lambda_k(t)\exp(\beta_k^T X_i)\right)/r_k\right)^{\sum_{k=1}^K I(\delta_i=k)+\theta^{-1}}} & r>0 \end{cases}$$

### 3.8.2 Derivation for Time to Multiple events

$$
\begin{aligned}
P(T=t,\delta_i=k|\omega_i,X_i) &= \int_0^\infty p(T=t,\delta_i=k|\omega_i,X_i)p(\omega_i|\theta,\alpha)d\omega_i \\
&= \int_0^\infty \prod_{k=1}^{K}\left[\omega_i\lambda_k(t)\exp\left\{\beta_k^T X_i\right\}\right]^{I(\delta=k)}\exp\left\{-\omega_i\Lambda_k(t)e^{\beta_k^T X_i}\right\} \\
&\quad \times\frac{\theta^\alpha}{\Gamma(\alpha)}\omega_i^{\alpha-1}e^{-\theta\omega_i}d\omega_i \\
&= \int_0^\infty \omega_i^{\sum_{k=1}^K I(\delta_i=k)}e^{-\omega_i\sum_{k=1}^K\Lambda_k(t)e^{\beta_k^T X_i}}\frac{\theta^\alpha}{\Gamma(\alpha)}\omega_i^{\alpha-1}e^{-\theta\omega_i}d\omega_i \\
&\quad \times\prod_{k=1}^{K}\left[\lambda_k(t)\exp\left\{\beta_k^T X_i\right\}\right]^{I(\delta_i=k)} \\
&= \int_0^\infty \omega_i^{\sum_{k=1}^K I(\delta_i=k)}e^{-\omega_i\sum_{k=1}^K\Lambda_k(t)e^{\beta_k^T X_i}}\frac{\theta^\alpha}{\Gamma(\alpha)}\omega_i^{\alpha-1}e^{-\theta\omega_i}d\omega_i \\
&\quad \times\prod_{k=1}^{K}\left[\lambda_k(t)\exp\left\{\beta_k^T X_i\right\}\right]^{I(\delta_i=k)} \\
&= \int_0^\infty \omega_i^{\sum_{k=1}^K I(\delta_i=k)+\alpha-1}e^{-\omega_i\left(\sum_{k=1}^K\Lambda_k(t)e^{\beta_k^T X_i}+\theta\right)}d\omega_i \\
&\quad \times\frac{\theta^\alpha}{\Gamma(\alpha)}\prod_{k=1}^{K}\left[\lambda_k(t)\exp\left\{\beta_k^T X_i\right\}\right]^{I(\delta_i=k)} \\
&= \frac{\Gamma\left(\alpha+\sum_{k=1}^K I(\delta_i=k)\right)}{\Gamma(\alpha)}\frac{\prod_{k=1}^K\left[\lambda_k(t)\exp\left\{\beta_k^T X_i\right\}\right]^{I(\delta_i=k)}\theta^\alpha}{\left(\sum_{k=1}^K\Lambda_k(t)e^{\beta_k^T X_i}+\theta\right)^{\alpha+\sum_{k=1}^K I(\delta_i=k)}} \\
&= \frac{\prod_{k=1}^K\left[\lambda_k(t)\exp\left\{\beta_k^T X_i\right\}\right]^{I(\delta_i=k)}\theta^\alpha}{\left(\sum_{k=1}^K\Lambda_k(t)e^{\beta_k^T X_i}+\theta\right)^{\alpha+\sum_{k=1}^K I(\delta_i=k)}}\prod_{j=1}^{\sum_{k=1}^K I(\delta_i=k)}(\alpha+j-1)
\end{aligned}
$$

- Further, using the spline basis formulation of the baseline hazard, we obtain

$$
\begin{aligned}
P(T=t,\delta_i=k|\omega_i,X_i) &= \frac{\prod_{k=1}^{K}\left[\lambda_k(t)\exp\left\{\beta_k^T X_i\right\}\right]^{I(\delta_i=k)}\theta^{\alpha}}{\left(\sum_{k=1}^{K}\Lambda_k(t)e^{\beta_k^T X_i}+\theta\right)^{\alpha+\sum_{k=1}^{K}I(\delta_i=k)}}\prod_{j=1}^{\sum_{k=1}^{K}I(\delta_i=k)}(\alpha+j-1) \\
&= \frac{\prod_{k=1}^{K}\left[\sum_k \alpha_k B_{k,q}(t)\exp\left\{\beta_k^T X_i\right\}\right]^{I(\delta_i=k)}\theta^{\alpha}}{\left(\sum_{k=1}^{K}\sum_{k=1}^{s-1}\left(\sum_{j=1}^{k}\alpha_j(\xi_{j+q}-\xi_j)/q\right)B_{k,q+1}(t)e^{\beta_k^T X_i}+\theta\right)^{\alpha+\sum_{k=1}^{K}I(\delta_i=k)}} \\
&\quad \times \prod_{j=1}^{\sum_{k=1}^{K}I(\delta_i=k)}(\alpha+j-1)
\end{aligned}
$$

### 3.8.3 Simulation Details

For the simulation studies, we generate data from non-proportional hazards model. Simulation strategies for proportional hazards may be found in (Bender et al., 2005) and we extend this methodology to the general transformation model. For each subject $i$ we first draw a frailty term $\omega_i \sim \Gamma(\theta^{-1},\theta^{-1})$. Next, conditional on $\omega_i$, we use the standard probability integral transformation generate the survival times. Notably, we simulate from

$$
T_{ik}^{*} = \left[\frac{\exp\left(\frac{\log(U)r}{-\omega_i}\right)-1}{\lambda_k r\exp\left(X_i'\beta_k\right)}\right]^{1/\rho_k}
$$

Since $\Lambda_k$ is an invertible map due to the fact that $\lambda_k(t) > 0$ for all values of $t$ and $k$. Once have simulated a set of $k$ survival times for each individual, we generate censoring times according to $C_i = 1 + Uc$ where $U \sim [0,1]$ and $c$ is some constant. Then, for the $i^{th}$ individual we take $T_{ik} = \min\{T_{ik}^{*}, C_i\}$. Finally, to simulate observing the first event, we obtain $\min\{T_{ik}\}_{k=1}^{K}$.

### 3.8.4 Univariate Cox Models

|  | Est. | HR | SE | z | p | Lower 95% CI | Upper 95% CI |
|---|---|---|---|---|---|---|---|
| genderMale | 0.317 | 1.373 | 0.020 | 15.652 | 0.000 | 0.277 | 0.356 |
| masthma | 0.270 | 1.310 | 0.043 | 6.210 | 0.000 | 0.185 | 0.355 |
| mat-smoke | 0.237 | 1.267 | 0.023 | 10.139 | 0.000 | 0.191 | 0.282 |
| assiss | -0.050 | 0.952 | 0.041 | -1.208 | 0.227 | -0.130 | 0.031 |
| c-section | 0.037 | 1.038 | 0.024 | 1.561 | 0.119 | -0.010 | 0.084 |
| AfAm | 0.027 | 1.027 | 0.026 | 1.021 | 0.307 | -0.025 | 0.078 |
| residence2 | 0.074 | 1.077 | 0.029 | 2.551 | 0.011 | 0.017 | 0.131 |
| resresidence3 | 0.390 | 1.477 | 0.028 | 13.883 | 0.000 | 0.335 | 0.445 |
| weight-stand | -0.084 | 0.920 | 0.010 | -8.020 | 0.000 | -0.104 | -0.063 |
| siblings | 0.008 | 1.008 | 0.008 | 1.030 | 0.303 | -0.007 | 0.023 |

Table 3.7: Univariate Cox PH model for emergency room admitted infant bronchiolitis

|  | Est. | HR | SE | z | p | Lower 95% CI | Upper 95% CI |
|---|---|---|---|---|---|---|---|
| genderMale | 0.286 | 1.331 | 0.015 | 18.574 | 0.000 | 0.256 | 0.316 |
| masthma | 0.059 | 1.061 | 0.035 | 1.692 | 0.091 | -0.009 | 0.128 |
| mat-smoke | 0.050 | 1.052 | 0.017 | 2.966 | 0.003 | 0.017 | 0.083 |
| assiss | 0.018 | 1.018 | 0.029 | 0.605 | 0.545 | -0.040 | 0.075 |
| c-section | 0.052 | 1.054 | 0.018 | 2.915 | 0.004 | 0.017 | 0.088 |
| AfAm | -0.382 | 0.683 | 0.022 | -17.680 | 0.000 | -0.424 | -0.339 |
| residence2 | 0.019 | 1.020 | 0.018 | 1.062 | 0.288 | -0.016 | 0.055 |
| resresidence3 | -0.438 | 0.645 | 0.022 | -19.861 | 0.000 | -0.481 | -0.395 |
| weight-stand | -0.046 | 0.955 | 0.008 | -5.941 | 0.000 | -0.062 | -0.031 |
| siblings | 0.015 | 1.015 | 0.006 | 2.408 | 0.016 | 0.003 | 0.027 |

Table 3.8: Univariate Cox PH model for outpatient clinic admitted infant bronchiolitis

|  | Est. | HR | SE | z | p | Lower 95% CI | Upper 95% CI |
|---|---|---|---|---|---|---|---|
| genderMale | 0.256 | 1.292 | 0.025 | 10.143 | 0.000 | 0.207 | 0.306 |
| masthma | 0.115 | 1.122 | 0.057 | 2.006 | 0.045 | 0.003 | 0.227 |
| mat-smoke | 0.207 | 1.230 | 0.027 | 7.570 | 0.000 | 0.153 | 0.261 |
| assiss | -0.019 | 0.981 | 0.050 | -0.386 | 0.699 | -0.117 | 0.078 |
| c-section | 0.068 | 1.070 | 0.029 | 2.304 | 0.021 | 0.010 | 0.126 |
| AfAm | -0.341 | 0.711 | 0.036 | -9.550 | 0.000 | -0.412 | -0.271 |
| residence2 | -0.383 | 0.681 | 0.032 | -12.009 | 0.000 | -0.446 | -0.321 |
| resresidence3 | -0.559 | 0.572 | 0.036 | -15.722 | 0.000 | -0.629 | -0.489 |
| weight-stand | -0.101 | 0.904 | 0.013 | -7.783 | 0.000 | -0.127 | -0.076 |
| siblings | 0.105 | 1.111 | 0.009 | 11.403 | 0.000 | 0.087 | 0.124 |

Table 3.9: Univariate Cox PH model for hospital admitted infant bronchiolitis

# CHAPTER 4

# SOFTWARE IMPLEMENTATION: THE BAYLEAF MODULE

## 4.1 Introduction

The implementation of statistical methodologies into usable tools for researchers and practitioners is paramount to a method's widespread adoption. The development of statistical software for general survival analysis has a strong presence in the R programming language environment (R Core Team, 2014). However, the implementation of survival analysis, especially multivariate survival analysis, in the python programming language (Python Core Team, 2015) is limited. The python module 'lifelines' offers standard univariate, frequentist inferential procedures including Cox proportional hazards and parametric survival analysis. Furthermore, univariate Bayesian models survival models may be found in the excellent 'survivalstan' package; in this module, standard Bayesian piecewise exponential models built upon the framework of Stan can be found. Despite the existence of these modules, software has not been implemented to perform Bayesian analysis of transformation-based frailty and copula models of the sort described previously in this dissertation. The utilization of the python programming language allows for tool development in one of the most commonly used data science languages. Additionally, the utilization of python allows usage of the deep-learning library theano for automatic computation of gradients for standard statistical procedures.

In this paper, we describe a new software module built specifically for the implementation of Bayesian transformation models developed and studied in this dissertation.

### 4.1.1 Goals and Vision

The goal of the bayleaf module is to provide an “easy to use” API to fit Bayesian transformation models using PyMC3 as a computational backend. By using PyMC3, we are able to build statistical models that can use next-generation sampling techniques as well as variational Bayesian methodologies for large datasets. Bayesian survival models have been historically limited in their wide-spread usability due to the limitations of previous software packages. Notably, Bayesian survival analysis has often been formulated in terms of the so called “Poisson trick”, wherein an intensity process is modeled and a Poisson likelihood is assumed. Using tools like PyMC3, we are instead able to circumvent the re-posing of the likelihood and are able to use log likelihoods in terms of any hazard and survival functions.

This version of the bayleaf package provides a user friendly syntax and modeling suite for the parametric models contained in this dissertation. This paper is organized in the following manner. First, we will describe the basic structure of the module (note, all of the code used to generate this dissertation is available on the author’s github). Next, we describe the different models currently contained in the module. For each, we provide an example and demonstration for usage. In conclusion, we finish the paper with a brief note on future directions.

## 4.2 Tensor-based computation

Before we discuss the inner-workings of the package, we briefly discuss the abstract way in which computation is undertaken in the package. The basis for the computation back-end of the bayleaf module is a graphical model. Specifically, a computational graph is simply a directed graph whose nodes are *operations* and whose edges are *tensors* (Abadi et al., 2016). We will include a directed acyclic graph for each model in their corresponding subsection for the user to visualize the computational process.

## 4.3 Installation

The module bayleaf found on the author's github page.

## 4.4 Simulation Routines

The package *bayleaf* provides data simulation routines for univariate, frailty, and limited copula models. Notably the following functions are available to simulate data:

```
# Weibull Simulation routine
sim_Weibull(N, lam, rho, beta, rateC, maxtime)
# multivariate generalized weibull model
sim_weibull_frail_generalized(betas, theta, X, lam, r, rho, maxtime, cens_end, n, k, first = False)
```

Details for the simulation routines for these models can be found in the Appendix of the first paper in this dissertation.

## 4.5 Models

### 4.5.1 Class Structure of Package

The general class organization of the bayleaf module is adapted from the GLM submodule contained in PyMC3. In general, three component classes comprise the overall construction. First, the independent component constructs and adds the general covariate information to the model context. Next, the general modeling class uses a specified likelihood function to undergo inference.

## 4.6 Syntax

In this section, we describe the syntax of the 'bayleaf' package. In the main model class for each inferential model, there is a class method that parses an R-like model call. The syntax of this framework is built upon the 'patsy' python module; a framework to describe statistical models. In essence, the 'patsy' package parses the formula passed to the constructer and creates the appropriate design matrix and outcome vectors. Finally, the likelihood is added to the model context and is able to used for Bayesian fitting.

For both the copula and frailty models, the outcome and independent variables are separated by a tilde ("∼"). Furthermore, the formulas are represented in a string. Hence, a formula used in this package will be of the form:

```
formula = outcome ~ independent
```

Furthermore, each component of the formula uses names within the dataset to construct the model. Linear specifications of the independent component use "+" operators and the outcome uses the following parsing pattern:

```
’([time_1, time_2,..., time_k],[delta_1, delta_2,..., delta_k])~X_1+X_2+...+X_p-1’
```

Naturally, this formula is parsed using base python functions. Once the data is stripped from the inputted data frame, the outcome variables and design matrices are then used to populate the instantiated tensor variables used in the rest of the computational graph. The following class method is used for this process:

```
 @classmethod
def from_formula(cls, formula, data, minibatch = False, priors=None,
                   vars=None, name=’’, model=None):
       import patsy
       outcomes= formula.split("~")[0]
       # get time variables
       time_vars = [v.strip() for v in outcomes[outcomes.find("([")+2:outcomes.find("]")].split(",")]
       #get event times
       event_raw = outcomes[outcomes.find("],")+2:]
       event_vars = [v.strip() for v in event_raw[event_raw.find("[")+1:event_raw.find("])")].split(",")]
       # Now get x, times, and events
       x = patsy.dmatrix(formula.split("~")[1].strip(), data)
       time = data[time_vars].as_matrix()
       event = data[event_vars].as_matrix()
       labels = x.design_info.column_names
       # add the data tensors to the computational graph
       x_tensor = theano.shared(np.asarray(x)+0., borrow = True)
       time_tensor = theano.shared(time+0., borrow = True)
```

```
        event_tensor = theano.shared(event+0., borrow = True)

        return cls(x=x_tensor, time=time_tensor, event=event_tensor, minibatch=minibatch, labels=labels,
                   priors=priors, vars=vars, name=name, model=model)
```

We can see that the class method takes the string literal formula and first parses which values correspond to the outcome variable. Next, the ‘patsy‘ package creates the corresponding design matrix according to the specification given in the formula syntax. Finally, the class method takes the created numpy arrays and converts them into tensor shared variables to be used within theano for later use in the PyMC3 engine.

### 4.6.1 Univariate Models

The bayleaf package provides statistical inference for several univariate models. These include the proportional hazards model and general univariate versions of the transformation models studied in this dissertation.

### 4.6.2 Transformation Frailty Models

Frailty models assume that the joint survival function is constructed conditionally upon a subject specific random effect. The model likelihood is given by

$$L(\boldsymbol{\theta}|X) = \begin{cases} \prod_{i=1}^{n} \dfrac{\theta^{-\sum_{k=1}^{K} I(\delta_i=k)}\theta^{-\theta^{-1}} \prod_{k=1}^{K}\left[\lambda_k(t)\exp(\beta_k^T X_i)\right]^{I(\delta_i=k)}}{\left(\sum_{k=1}^{K}\Lambda_k(t)\exp(\beta_k^T X_i)+\theta^{-1}\right)^{\theta^{-1}+\sum_{k=1}^{K} I(\delta_i=k)}} & r = 0 \\ \prod_{i=1}^{n} \dfrac{\theta^{-1\sum_{k=1}^{K} I(\delta_i=k)}\theta^{-\theta^{-1}} \prod_{k=1}^{K}\left[\frac{\lambda_k(t)\exp(\beta_k^T X_i)}{1+r\Lambda_k(t)\exp(\beta_k^T X_i)}\right]^{I(\delta_i=k)}}{\left(\theta^{-1}+\sum_{k=1}^{K}\log\left(1+r\Lambda_k(t)\exp(\beta_k^T X_i)\right)/r\right)^{\sum_{k=1}^{K} I(\delta_i=k)+\theta^{-1}}} & r > 0. \end{cases}$$

In bayleaf, the following default prior distributions are placed on the parameters:

```
default_regressor_prior = Normal.dist(mu=0, tau=1/100)
default_lambda_prior = Gamma.dist(0.001,0.001, testval = 1.)
default_rho_prior = Gamma.dist(0.001,0.001, testval = 1.)
default_r_prior = InverseGamma.dist(alpha =1., testval = 1.)
default_theta_prior = Gamma.dist(0.001,0.001, testval = 1.)
```

|  | Mean | SD | MC Error | $HPD_{2.5}$ | $HPD_{97.5}$ | $n_{eff}$ | $\hat{R}$ |
|---|---|---|---|---|---|---|---|
| $X_{10}$ | 0.640 | 0.029 | 0.000 | 0.584 | 0.698 | 5595.988 | 1.0 |
| $X_{20}$ | 0.030 | 0.001 | 0.000 | 0.027 | 0.033 | 3226.548 | 1.0 |
| $X_{11}$ | 0.833 | 0.030 | 0.000 | 0.772 | 0.890 | 4431.405 | 1.0 |
| $X_{21}$ | 0.029 | 0.001 | 0.000 | 0.026 | 0.032 | 3876.154 | 1.0 |
| $X_{12}$ | 0.889 | 0.028 | 0.000 | 0.835 | 0.945 | 4810.688 | 1.0 |
| $X_{22}$ | 0.030 | 0.001 | 0.000 | 0.028 | 0.033 | 4372.268 | 1.0 |
| $\lambda_0$ | 0.005 | 0.000 | 0.000 | 0.004 | 0.005 | 2818.672 | 1.0 |
| $\rho_0$ | 0.710 | 0.011 | 0.000 | 0.689 | 0.731 | 3424.207 | 1.0 |
| $r_0$ | 0.448 | 0.088 | 0.001 | 0.281 | 0.618 | 3568.984 | 1.0 |
| $\lambda_1$ | 0.004 | 0.000 | 0.000 | 0.003 | 0.004 | 2845.079 | 1.0 |
| $\rho_1$ | 0.716 | 0.011 | 0.000 | 0.697 | 0.738 | 3367.490 | 1.0 |
| $r_1$ | 0.445 | 0.088 | 0.001 | 0.278 | 0.619 | 3606.243 | 1.0 |
| $\lambda_2$ | 0.006 | 0.000 | 0.000 | 0.005 | 0.007 | 3468.263 | 1.0 |
| $\rho_2$ | 0.716 | 0.010 | 0.000 | 0.696 | 0.734 | 3970.470 | 1.0 |
| $r_2$ | 0.460 | 0.060 | 0.001 | 0.346 | 0.580 | 4335.616 | 1.0 |
| $\theta$ | 0.505 | 0.015 | 0.000 | 0.475 | 0.536 | 7550.863 | 1.0 |

Table 4.1: Results for MCMC Frailty model

#### *4.6.2.1 Example*

To illustrate the usage of this model, we first generate data from the transformation model and subsequently fit using the formula class method.

```
with pm.Model() as model_test2:
    Frailty.from_formula(’([time_1, time_2, time_3],[delta_1, delta_2, delta_3])~X_1+X_2-1’, data = data)
    MAP = pm.find_MAP()
```

```
with model_test2:
    trace_mcmc =pm.sample(2000, start=MAP)
```

After fitting this model, we use the ‘summary’ function to generate a table of results. Table 4.1 presents these results. These results are obtained via the PyMC3 summary function.

The corresponding PyMC3 model for this particular implementation is given by:

```
with Model() as ordinal_surv_weibull_transformation_newMC:
    ### works with any prior we place on this
    ## first define hyperparameter priors for weibull components
    rho = pm.Gamma(’rho’, 0.001,0.001, shape = 3)
    lam = pm.Gamma(’lam’, 0.001,0.001, shape = 3) # k dimensions
            ## define hyperparameters for gamma frailty
    theta = pm.Gamma(’theta’,0.001,0.001)
            ## create matrix of beta priors of shape kxp
    beta = pm.Normal(’beta’, np.zeros(10), np.ones(10)*100, shape = (3,10))
```

```
## transformation parameter, can model with an Inverse Gamma.
r = pm.Gamma("r",0.01,0.01, shape = (3,1)) # pm.InverseGamma("r", alpha =1, shape = (3,1))
## Now define different components of log likelihood
def logp(delta_full, X_full, tau_full):
    linear = tt.dot(beta,X_full.T).T# this is the correct formulation
    weib_base_haz = lam*rho*tau_full**(rho-1) #weib haz
    weib_base_cumhaz = lam*tau_full**(rho) # cumulative ha
    phi_1 = tt.log(weib_base_haz*np.exp(linear))
    phi_2 = tt.log((1+r*weib_base_cumhaz*np.exp(linear)))
    failed_component = tt.sum(delta_full*phi_1, axis = 1)-tt.sum(delta_full*phi_2, axis = 1)
    psi = tt.log(tt.sum(tt.log(1+r*weib_base_cumhaz*tt.exp(linear))/r,axis=1)+theta**(-1))
               # second component for all the censored observations
    one_k = tt.ones(3)
    second = (theta**(-1)+tt.dot(delta_full, one_k))*psi
    # define log likelihood
    return tt.log(theta**(-tt.dot(delta_full,one_k))) + failed_component + theta**(-1)*tt.log(theta**(-1)) - second

survival = pm.DensityDist('survival', logp, observed={'delta_full':delta_full,
                                                'tau_full': tau_full,
                                                'X_full': X_full},
                       total_size = X_full.shape[0])
```

### 4.6.3 Transformation Copula Models

Similarly to the frailty model discussed in the previous section, the class structure for the copula based estimation is based on a base constructor class that first parses the input variables as specified by the formula. The formula syntax is exactly the same as in the frailty model.

For a given copula, $C_\alpha$, our likelihood is given by

$$\begin{aligned}
L(\alpha, \beta_1, \beta_2, \lambda_1, \lambda_2, r_1, r_2) &= \prod_{i=1}^{n} P(T_1 = t_1, T_2 = t_2)^{\delta_1 \delta_2} P(T_1 > t_1, T_2 = t_2)^{(1-\delta_1)\delta_2} \\
&\quad \times P(T_1 = t_1, T_2 > t_2)^{(1-\delta_2)\delta_1} P(T_1 > t_1, T_2 > t_2)^{(1-\delta_1)(1-\delta_2)} \\
&= \prod_{i=1}^{n} (c_\alpha(S_{i1}(t_1|X), S_{i2}(t_2|X)) f_{i1}(t_1|X) f_{i2}(t_2|X))^{\delta_1 \delta_2} \\
&\quad \times \left( -\frac{\partial C_\alpha(S_{i1}(t_1|X), S_{i2}(t_2|X)}{\partial S_{i1}(t_1|X)} \cdot (-f_{i1}(t_1|X)) \right)^{\delta_{i1}(1-\delta_{i2})} \\
&\quad \times \left( -\frac{\partial C_\alpha(S_{i1}(t_1|X), S_{i2}(t_2|X)}{\partial S_{i2}(t_2|X)} \cdot (-f_{i2}(t_2|X)) \right)^{\delta_{i2}(1-\delta_{i1})} \\
&\quad \times C_\alpha(S_{i1}(t_1|X), S_{i2}(t_2|X))^{(1-\delta_1)(1-\delta_2)}. \qquad (4.1)
\end{aligned}$$

as developed in the first paper of this dissertation. We use the following prior distributions on the model parameters:

```
 priors = {’alpha’:pm_dists.HalfCauchy.dist(beta=5),
            ’lam_1’: pm_dists.HalfCauchy.dist(beta=2.5),
            ’rho_1’: pm_dists.HalfCauchy.dist(beta=2.5),
           ’lam_2’: pm_dists.HalfCauchy.dist(beta=2.5),
           ’rho_2’: pm_dists.HalfCauchy.dist(beta=2.5),
            ’r_1’:pm_dists.HalfCauchy.dist(beta=2.5),
            ’r_2’:pm_dists.HalfCauchy.dist(beta=2.5)}
```

#### *4.6.3.1 Example*

To demonstrate this function, we first simulate data, then we initialize the model with 1500 tuning steps and run for 1000 samples. Results for the MCMC are presented in table 4.2. We used the following parameters for the simulation:

```
beta1 = np.array([1.63, 0.03])
beta2 = np.array([0.8, 0.03])
lambda_k = np.array([0.0047, 0.0037])
rho_k = np.array([0.716, .725])
alpha = 3.210
r_s = np.array([0.5,1.5])
```

```
with pm.Model() as test_run:
    Copula(time_1=time_1, time_2=time_2, e_1=delta_1, e_2=delta_2, x=X, family = ’clayton_trans’)
    MAP = pm.find_MAP()
    trace = pm.sample(1000, tune =1500, start = MAP)
```

Alternatively, we can use the formula syntax:

```
with pm.Model() as test_run:
    Copula.from_formula(’([time_1, time_2],[delta_1, delta_2])~X_1+X_2-1’, data = data, family = ’clayton_trans’)
    MAP = pm.find_MAP()
    trace = pm.sample(1000, tune =1500, start = MAP)
```

Table 4.2 contains the results of the MCMC run for the simulated dataset.

## 4.7 Future Directions

The ‘bayleaf module is still in its infancy and we are excited for what the future holds. Future developments for the ‘bayleaf’ package include the inclusion of B-splines in baselines of both frailty and copula models. Additionally, copula models can be numerically unstable in both frequentist and Bayesian frameworks, especially at the initialization stage of MCMC; therefore, more robust initialization routines will be

|  | Mean | SD | MC Error | $HPD_{2.5}$ | $HPD_{97.5}$ | $n_{eff}$ | $\hat{R}$ |
|---|---|---|---|---|---|---|---|
| $X_{11}$ | 1.686 | 0.035 | 0.001 | 1.622 | 1.756 | 1597.371 | 1.000 |
| $X_{21}$ | 0.032 | 0.001 | 0.000 | 0.030 | 0.034 | 1739.013 | 1.000 |
| $X_{12}$ | 0.826 | 0.040 | 0.001 | 0.748 | 0.905 | 2071.032 | 1.000 |
| $X_{22}$ | 0.033 | 0.002 | 0.000 | 0.029 | 0.036 | 1749.652 | 1.000 |
| $\rho_1$ | 0.723 | 0.011 | 0.000 | 0.701 | 0.742 | 1474.589 | 1.000 |
| $\rho_2$ | 0.752 | 0.015 | 0.000 | 0.720 | 0.779 | 1765.541 | 1.003 |
| $\lambda_1$ | 0.004 | 0.000 | 0.000 | 0.004 | 0.005 | 1342.057 | 1.000 |
| $\lambda_2$ | 0.003 | 0.000 | 0.000 | 0.002 | 0.004 | 1618.397 | 1.001 |
| $\alpha$ | 3.212 | 0.067 | 0.001 | 3.082 | 3.343 | 3267.176 | 1.000 |
| $r_1$ | 0.504 | 0.037 | 0.001 | 0.430 | 0.574 | 1813.504 | 1.000 |
| $r_2$ | 1.578 | 0.109 | 0.003 | 1.357 | 1.787 | 1621.922 | 1.003 |

Table 4.2: Results for MCMC Copula model

added to the framework in future releases (see the first paper of this dissertation for a discussion of this). To accomplish these tasks, we will have separate class specifications for the outcome and independent components and include separate routines to incorporate univariate marginal estimates as initial values. We also hope to extend the bivariate copula models to be of arbitrary dimension. Naturally, we would like to include differing correlation structures between pairwise dimensions, so a conditional specification is most likely.

As of the writing of this dissertation, the deep-learning library ‘theano’ is no longer actively developed (although it is maintained for its current version). As the PyMC project moves forward (which will possibly involve switch of computational backend to another deep-learning library), we will be adapting our models accordingly.

# CHAPTER 5

# CONCLUSION

In this dissertation, we have developed and explored Bayesian tools for generalized multivariate survival models. In the first portion of this thesis, we focused on generalizing various bivariate copula survival models that allow for separate specification of marginal and association components. In this section, we generalized several preexisting Archimedean copula models to include more flexible forms of the underlying marginal models. Specifically, we generalized the structure of the underlying cumulative hazard in each margin to include common cases of survival models including the proportional hazards and proportional odds models. We learned this generalization as a transformation parameter included as part of the overall Bayesian solution. First, we studied survival models that include parametric baseline hazards; however, we provided an additional source of flexibility by introducing a semi-parametric specification through a B-spline function for the baseline hazard. For the semi-parametric model, we studied both penalized and un-penalized approaches and found that penalizing the spline coefficients through the introduction of a hierarchical parameter provided better overall frequentist coverage over the entirety of the parameter space. The semi-parametric model inherently can model any shape of baseline hazard, thus removing a restrictive assumption of parametric models. In principle, the penalization of spline coefficients may allow for the usage of a large number of knots in the spline specification. We found during the course of our simulations that using a large number of spline knots in tandem with the initialization process has the ability to produce errors in the initialization procedure when too many knots are introduced due to the increase in dimensionality of the parameter space for optimization. A two stage initialization is recommended for both parametric and semi-parametric models for computational stability.

In the second portion of this thesis, we developed a scalable Bayesian framework to accommodate time to first event of multivariate survival outcomes with ordinal severity. This is done using a flexible Bayesian multivariate frailty model that de-restricts the form of the survival function in order to simultaneously study the correlated covariate effects on differing severity levels of the outcome and to provide a mechanism for combining these profiles into an overall effect. This work was motivated by and applied to the Tennessee Asthma Bronchiolitis Study (TABS) cohort (derived from the TennCare medical claims database) in order to quantify maternal smoking effect across levels of first hospital admittance for infant bronchiolitis. Using an additional data source correlating the multivariate survival outcomes with ordinal severity scores, we provide a systematic and flexible way to determine the overall direction of the smoking effect size over the multivariate survival events. Furthermore, we investigated the effect of using a scalable Bayesian algorithmic framework to perform inference in large datasets. We found that for time to first event data, the usage of a frailty transformation model is efficacious for finding the effect sizes and baseline hazard components. Due to the increase of within-person censoring, however, we found that the simultaneous determination of the transformation parameter and the frailty parameter to be difficult. Specifically, fixing the transformation parameter at the *maximum a posteriori* value produced posterior distributions for the frailty parameter to be too narrow, and thus did not capture the true parametric value in simulated results. We confirmed this notion by running the same simulation in time to any event, where multiple events can be observed for each person.

For a final section of this dissertation, we provided software for the previously described time-to-event methods using the python Bayesian module PyMC3 which is built upon the deep learning library ‘theano’. This software includes flexible implementations of the methods developed in this dissertation with user-friendly syntax to reduce the barrier-of-entry to researchers. Leveraging PyMC3 allows for usage of the deep learning library to automatically compute otherwise difficult quantities such as gradient information necessary for Bayesian methods as well as Hybrid Monte Carlo sampling and variational inference.

## 5.1 Future Research

The continued influx of vast medical data provides a foundation for an exciting future of medical research. Specifically, increased amounts of sample information allow us to develop richer and more flexible modeling capabilities. Furthermore, richer models allow us to make better decisions and predictions under uncertainty. In my view, the Bayesian approach to deep learning is an exciting and challenging domain with much promise. By constructing neural-network architectures with probabilistic weights, one can quantify uncertainty in predicted outputs more rigorously than in non-Bayesian networks of the same kind. Additionally, such frameworks may allow us to make optimal decisions under uncertainty by leveraging the decision-theoretic framework inherent in Bayesian methods. With the tools we have developed and explored in this dissertation, as well as the development of methods for integrating heterogeneous data sources, one tangible goal of future research would be integrating complex deep architectures within multivariate time-to-event analyses. While constructing such models in theory may be straightforward, what is still unknown is the efficacy of Variational Bayesian methods in general for Bayesian neural networks. It is known that variational Bayes has the ability to under-estimate the variance in the posterior distributions. Hence, it is important for the development of any system that utilizes Variational Bayesian methods to see how sensitive the posterior is to various distributional assumptions contained within the objective function. Underestimation of variance can lead to incorrect inference and quantification of a patient's risk. In a hospital setting, underestimating variability can also result in monetary loss, especially if decisions are being made within the Bayesian framework (Graves, 2011).

Continuing in the vein of Bayesian Neural networks, it has been shown that an infinitely wide deep neural network is equivalent to a Gaussian Process (Lee et al., 2017). While this dissertation has had a secondary focus on scalable inference, it is also important to consider increased flexibility in smaller datasets. Specifically, the investigation of Gaussian process priors (Rasmussen, 2004) in the setting of survival analysis is relatively recent development; key references can be found in Fernández et al. (2016). These models posit a zero-mean Gaussian process prior for functional forms. We envision the extension of our proposed methods to be able to accommodate the inclusion of Gaussian process priors for non-linear independent components. Figure 5.1 shows a posterior fit for an intensity process for univariate survival model.

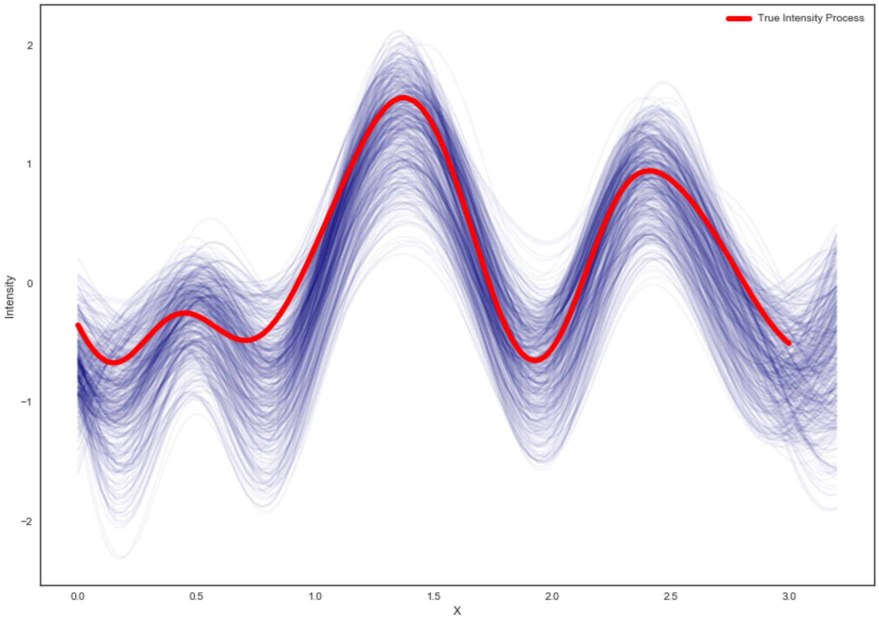


Figure 5.1: Posterior MCMC draws for intensity process for Gaussian process survival

The adaptation of Bayesian methods at scale is quite challenging as we have seen in this dissertation. Primarily, the restriction of variational methods to Gaussian distributional families in turn restricts posterior inference through the introduction of further assumptions. While adapting the variational approach to neural network based models, it is important to de-restrict the approximating distribution to allow for bi-modal and other general shapes of the posterior. One predominant approach to this flexibility is the introduction of Normalizing flows (Rezende and Mohamed, 2015), wherein the shape of the posterior is inferred as part of the optimization itself. Specifically, the normalizing flows procedure formulates the approximating distributional family as a series of invertible transformations ($f_k$) of a standard normal distribution, $z_0$; see equations 5.1 and 5.2. This particular approach does not come with out its challenges, however. Specifically, during the course of fitting, it is especially important to track the derivatives alongside the model parameters for each step in the fitting process.

$$z_K = f_K \circ \cdots \circ f_2 \circ f_1(z_0) \tag{5.1}$$

$$\ln q_K(z_K) = \ln q_0(z_0) - \sum_{k=1}^{K} \ln \left| \frac{\partial f_k}{\partial z_{k-1}} \right| \tag{5.2}$$

One pervasive problem in the area of normalizing flows is a lack of understanding in which and how many flows are appropriate in a given statistical scenario; this problem is similar to the choice of activation functions for a neural network. The appropriate transformations for normalizing flows variational inference has not been studied, to our knowledge, in the setting of multivariate survival analysis. Additionally, while there have been some theoretical guarantees regarding posterior concentration of the variational approach to general Bayesian problems (Wang and Titterington, 2012), this has not been thoroughly studied with respect to normalizing flows framework.

In addition to the previously developed parametric framework for the frailty model for time-to-first event, we are currently implementing a semi-parametric version of the present model using the spline specification developed in the second paper of this thesis. Specifically, we replace the baseline hazard function $\lambda_0(t)$ with a linear constraint given by

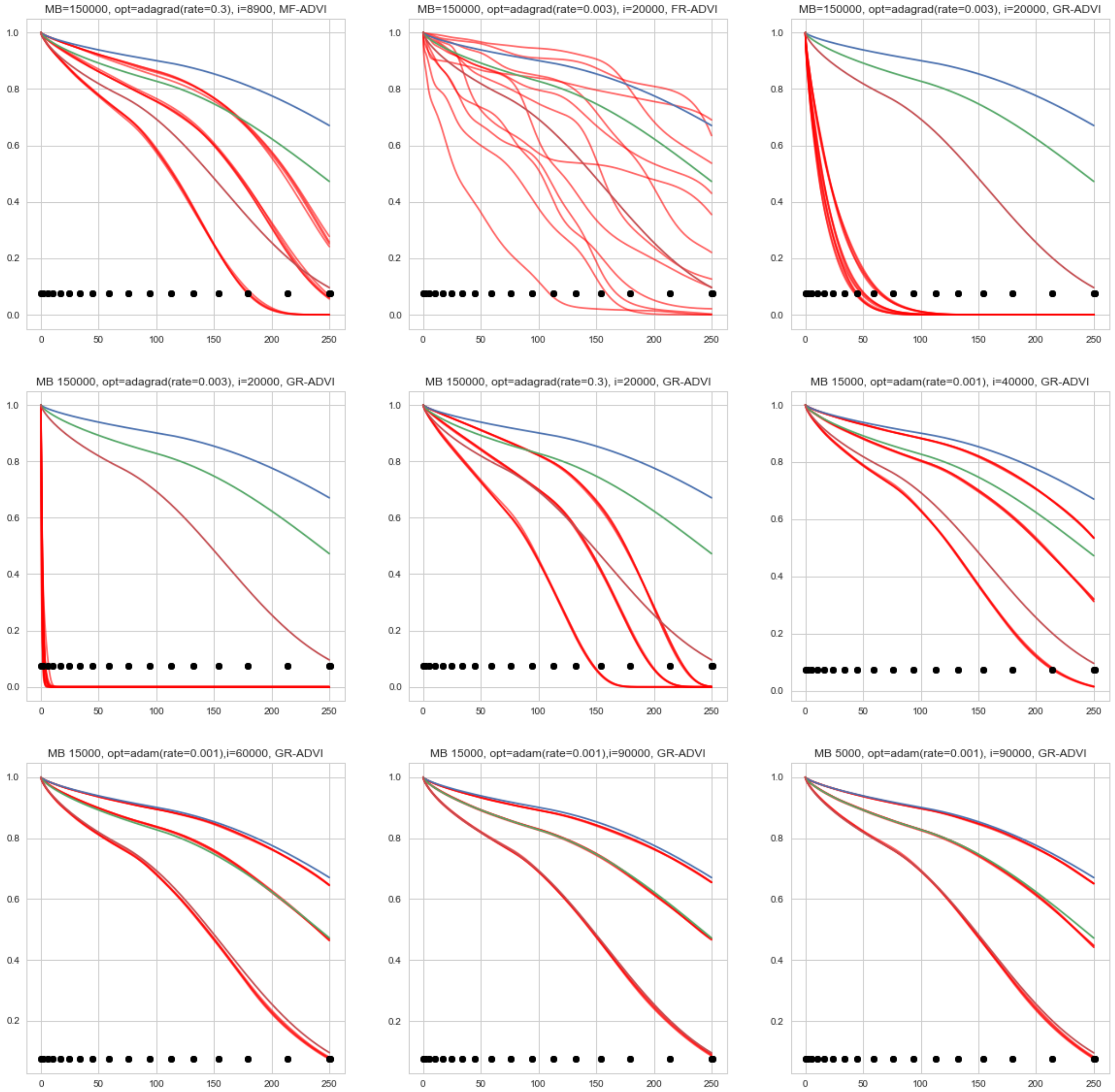
$$\lambda_0(t) \quad = \quad \gamma_0 + \gamma_1 t + \sum_k \alpha_k B_{k,q}(t)$$

which implies that the cumulative hazard is given by

$$\begin{aligned}
\Lambda_0(t) &= \int_0^t \lambda_0(x)dx \\
&= \int_0^t \gamma_0 + \gamma_1 x + \sum_k \alpha_k B_{k,q}(x)dx \\
&= \frac{1}{2}t(2(\gamma_0 + \gamma_1 t)) + \int_0^t \sum_k \alpha_k B_{k,q}(x)dx \\
&= \frac{1}{2}t(2(\gamma_0 + \gamma_1 t)) + \sum_{k=1}^{s-1} \left( \sum_{j=1}^{k} \alpha_j (\xi_{j+q} - \xi_j)/q \right) B_{k,q+1}(t).
\end{aligned}$$

The likelihood specification follows from before in equation (3.8). A semi-parametric version of the aforementioned model would provide a flexible alternative to modeling the joint survival function with the restrictive Weibull model. We found during the course of our experimentation that many factors are necessary to identify when implementing ADVI with a spline baseline hazard. As described, we take the same approach to implementing the spline-baseline hazard model as in the first paper of this dissertation. For this approach, 20 knots were placed at evenly spaced sample quantiles. One main challenge with this approach is determining the best variational inferential settings for proper inference of a given model. Figure 5.2 illustrates several combinations of settings for VI for this spline model.

One final future point of research is the extension of the methods developed in this thesis to arbitrary dimensions. It should be noted that although the copula methods contained herein can be extended to $n$ dimensions, the main assumption of the copula model is that the pairwise dependence parameter between each dimension


MB=150000, opt=adagrad(rate=0.3), i=8900, MF-ADVI

MB=150000, opt=adagrad(rate=0.003), i=20000, FR-ADVI

MB=150000, opt=adagrad(rate=0.003), i=20000, GR-ADVI

MB 150000, opt=adagrad(rate=0.003), i=20000, GR-ADVI

MB 150000, opt=adagrad(rate=0.3), i=20000, GR-ADVI

MB 15000, opt=adam(rate=0.001), i=40000, GR-ADVI

MB 15000, opt=adam(rate=0.001),i=60000, GR-ADVI

MB 15000, opt=adam(rate=0.001),i=90000, GR-ADVI

MB 5000, opt=adam(rate=0.001), i=90000, GR-ADVI

Figure 5.2: Differing Combinations of Optimization Criteria for ADVI in Semiparametric Model

is the same. Unless we expect each dimension of the outcome to covary identically, then one must develop ways to develop a multivariate model with differing dependence parameters for each dimension pair. The Vine method of developing graphical models Bedford and Cooke (2002), paired with the copula methodology can enable higher dimensional, flexible modeling through the introduction of transformation models and semi-parametric baseline hazards developed in this dissertation. Additionally, a graphical approach to computing is beneficial here for programming. The ability to modularize programming tasks can more easily facilitate the Bayesian learning for graphical models like Vine copulas; conditional on the assumed dependence structure between the variety of dimensions, one can specify a likelihood function in a piecewise manner. Additionally, learning the underlying conditioning may be possible in this framework as well.

As for the development of the software package ‘bayleaf’ we are in the process of implementing a computational routine for the spline models. One particular challenge in software development is the appropriate default settings for variational Bayesian inference for spline-based estimation. As we saw, finding optimal settings for the various tuning parameters is indeed challenging. However, for the general MCMC case, we will work towards making our models more flexible and robust.

## 5.2 Appendix

### 5.2.1 Code for Paper 1

#### *5.2.1.1 Code for Semi-Parametric Copula Model*

```
import pymc3 as pm
import numpy as np
import pandas as pd
import theano
import statsmodels.api as sm
import theano.tensor as tt
import patsy
from patsy import dmatrix
import numpy as np
from scipy import special
import scipy as sp
import pandas as pd

def knot_matrix(orde, xi, time):
    #orde = 4 ## order of the spline
    ## in the augmented knot matrix, you need to repeat the upper bound and lower bound orde times each
    knot_augment = np.append(np.append(np.repeat(0.,repeats=orde), xi), np.repeat(max(time)+2,repeats=orde))
    nk = len(knot_augment)
    knot_aug2 = np.append(knot_augment, knot_augment[nk-1])
    foo = np.append(((knot_aug2[(orde):]-knot_aug2[:-(orde)])/float(orde))[:-2],[0])
    diff_mat = np.array([foo for i in range(len(foo)-1)])
    masked_k = np.reshape(np.multiply(np.tri(N=diff_mat.shape[0],
                                             M = diff_mat.shape[1]).flatten(),
                                      diff_mat.flatten()),
                          diff_mat.shape)
    aug_masked_k = np.row_stack([np.zeros(len(foo)),masked_k]).T
    return(aug_masked_k.tolist())
# 3. We'll do order 4 spline bases (i.e. cubic splines)
def bh_basis_fxns(time, xi):
    import patsy
    fxns = np.asarray(patsy.dmatrix("bs(x, knots = xi, lower_bound = 0., upper_bound = max(time)+0.00000000001,\
                                    degree = 3, include_intercept=True) - 1",\
                                    {"x": time})).tolist()
    return(fxns)
# To play nicely, we need to lop off the last element
def cum_bh_fxns(time, xi):
    import patsy
    fxns = np.asarray(patsy.dmatrix("bs(x, knots = xi, lower_bound = 0., upper_bound = max(time)+0.00000000001,\
                                    degree = 4, include_intercept=True) - 1",\
                                    {"x": time}))[:,:-1].tolist()
    return(fxns)
## Covariate Generation
def sim_simple_covs(n):
    sex = np.random.binomial(n=1,p=.5,size =n)
    age = np.random.gamma(size=n, shape = 10, scale = 1/.3)
    return(np.array([sex,age]).T)
## Outcome generation, Weibull_BH
# Weird BH
def knot_matrix(orde, xi, time):
    #orde = 4 ## order of the spline
    ## in the augmented knot matrix, you need to repeat the upper bound and lower bound orde times each
    knot_augment = np.append(np.append(np.repeat(0.,repeats=orde), xi), np.repeat(max(time)+0.00000000001,repeats=orde))
    nk = len(knot_augment)
    knot_aug2 = np.append(knot_augment, knot_augment[nk-1])
    foo = np.append(((knot_aug2[(orde):]-knot_aug2[:-(orde)])/float(orde))[:-2],[0])
    diff_mat = np.array([foo for i in range(len(foo)-1)])
    masked_k = np.reshape(np.multiply(np.tri(N=diff_mat.shape[0],
                                         M = diff_mat.shape[1]).flatten(),
                                  diff_mat.flatten()),
                        diff_mat.shape)
    aug_masked_k = np.row_stack([np.zeros(len(foo)),masked_k]).T
    return(aug_masked_k.tolist())
# 3. We'll do order 4 spline bases (i.e. cubic splines)
def bh_basis_fxns(time, xi):
    import patsy
    fxns = np.asarray(patsy.dmatrix("bs(x, knots = xi, lower_bound = 0., upper_bound = max(time)+0.00000000001,\
                                 degree = 3, include_intercept=True) - 1",\
                                 {"x": time})).tolist()
    return(fxns)
# To play nicely, we need to lop off the last element
def cum_bh_fxns(time, xi):
    import patsy
    fxns = np.asarray(patsy.dmatrix("bs(x, knots = xi, lower_bound = 0., upper_bound = max(time)+0.00000000001,\
                                 degree = 4, include_intercept=True) - 1",\
                                 {"x": time}))[:,:-1].tolist()
    return(fxns)
## Covariate Generation
def sim_simple_covs(n):
    sex = np.random.binomial(n=1,p=.5,size =n)
    age = np.random.gamma(size=n, shape = 10, scale = 1/.3)
    return(np.array([sex,age]).T)
## Outcome generation
# First, generate an instance of data to init the graph

### We need to see how well the map does for the Weird BH
weib_bh = lambda rho, lam, t: rho*lam*t**(rho-1)
weib_cbh = lambda rho, lam, t: lam*t**(rho)
##Since we can invert parts of a piecewise function, we then have a way to perform the integration on this
def weird_bh1(t, rho1=.725, lam1=0.0037, rho2=2.714, lam2=0.0000001, changepoint =100):
    res = []
    for time in t:
        if time <= changepoint:
            res = np.append(res,weib_bh(rho = rho1, lam = lam1, t = time))
        elif (time > changepoint):
            res = np.append(res,weib_bh(rho = rho2, lam = lam2, t = time))
    return(res)
t = np.linspace(0.001,365, 100)
weib_bh = lambda rho, lam, t: rho*lam*t**(rho-1)

## cumulative hazard
def weird_cbh1(t, rho1=.725, lam1=0.0037, rho2=2.714, lam2=0.0000001, changepoint = 100):
    res = []
    for time in t:
        if time <= changepoint:
            integral = weib_cbh(rho1, lam1, time)
            res = np.append(res,integral)
        elif (time > changepoint):
            integral = weib_cbh(rho1, lam1, changepoint)+(weib_cbh(rho=rho2,lam=lam2,t=time)-weib_cbh(rho=rho2,lam=lam2,t = changepoint))
            res = np.append(res,integral)
    return(res)

## now to the simulate data functions
def weird_cbh1_inverse(u, rho1=.725, lam1=0.0037, rho2=2.714, lam2=0.0000001, changepoint=100):
    res = []
    for u_t in u:
        if u_t <= weird_cbh1([changepoint], rho1, lam1, rho2, lam2):
            inverse = (u_t/lam1)**(1/rho1)
            res = np.append(res, inverse)
        elif u_t > weird_cbh1([changepoint], rho1,lam1,rho2,lam2):
            inverse = ((u_t-weib_cbh(rho=rho1, lam=lam1, t=changepoint)+weib_cbh(rho=rho2,lam=lam2, t = changepoint))/lam2)**(1/rho2)
            res = np.append(res, inverse)
    return(res)

## Simulate from copula
n = 5000
beta1 = np.array([-0.63, 0.03])
beta2 = np.array([-0.69, 0.02])
betas = np.vstack([beta1,beta2])
betas = np.vstack([beta1,beta2])

n_tune = 2500 #number of tuning steps for mcmc


X = sim_simple_covs(n)
exp_1 = np.exp(np.dot(X,betas.T))[:,0]
exp_2 = np.exp(np.dot(X,betas.T))[:,1]

alpha = 3.210

r_s =0.5
r_t =0.5
# clayton generation
U_s = np.random.uniform(size=n)
S = (U_s**(-r_s)-1)/(r_s*exp_1)
# S trans
## Edit for sept 9, change to see whether we need a different bh
rhos = np.array([[.725, .725], [2.714, 2.714]])
lams = np.array([[0.0037, 0.0067], [0.0000001, 0.0000005]])
changepoints = np.array([100,70])

S_trans = weird_cbh1_inverse(u = S,rho1=rhos[0,0],lam1=lams[0,0],
                                rho2=rhos[1,0], lam2=lams[1,0],
                                changepoint=changepoints[0])
U_t = np.random.uniform(size=n)
S_T = ((U_t**(-(alpha)/(alpha+1))-1)*(U_s)**(-alpha)+1)**(-alpha**(-1))
T = (S_T**(-r_t)-1)/(r_t*exp_2)
T_trans = weird_cbh1_inverse(u = T,rho1=rhos[0,1],lam1=lams[0,1],
                                rho2=rhos[1,1], lam2=lams[1,1],
                                changepoint=changepoints[1])



Te = np.vstack([S_trans,T_trans]).T
cens_end = 7500
```

```
maxtime = 300
Cens = 1+cens_end*np.random.uniform(size = (n,2))
Cens[Cens>maxtime] = maxtime

Cens = 1+cens_end*np.random.uniform(size = (n,2))
Cens[Cens>maxtime] = maxtime
results = np.repeat(0, n)
names_df = ["del"]
# loop over levels
for level in range(2):
    obs_t = np.amin(np.array([Te[:,level], Cens[:,level]]).T, axis =1) # observed time
    names_df = np.append(names_df, "time_"+str(level+1))
    delta = (Te[:,level] < Cens[:,level]) + 0 # censoring indicator
    names_df = np.append(names_df, "delta_"+str(level+1))
    results = np.vstack((results, obs_t))
    results = pd.DataFrame(np.vstack((results, delta)))
#
x_names = ["X_"+str(j+1) for j in np.arange(X.shape[1])]
names_df = np.append(names_df, x_names)
#names_df = np.append(names_df, "frailty") # now add frailty
out = pd.DataFrame(np.vstack((results, X.T)).T)
out.columns = names_df
out = out.iloc[:, out.columns!="del"]

#### Now create data vectors and tensors
tau1_full = theano.shared(np.array(out["time_1"])+0.,borrow=True)
tau2_full = theano.shared(np.array(out["time_2"])+0.,borrow=True)
delta1_full = theano.shared(np.array(out["delta_1"])+0., borrow = True)
delta2_full = theano.shared(np.array(out["delta_2"])+0., borrow = True)

X_full = theano.shared(X+0., borrow = True)

tau1 = np.array(out["time_1"])+0.
tau2 = np.array(out["time_2"])+0.
delta1 = np.array(out["delta_1"])+0.
delta2 = np.array(out["delta_2"])+0.
###
#### Now construct the splines for each dimension
#### Dimension 1.

#times = np.sort(time)
## spline knots
knots_num = 15

t_obs = tau1[np.array(out["delta_1"], dtype =int)==1]
xi = np.append(np.percentile(t_obs, q = np.arange(0,100,100/((knots_num)-1))),[max(tau1)])
orde = 4 ## order of the spline
## in the augmented knot matrix, you need to repeat the upper bound and lower bound orde times each
knot_augment = np.append(np.append(np.repeat(0.,repeats=orde), xi), np.repeat(max(tau1)+0.00000000001,repeats=orde))
nk = len(knot_augment)
knot_aug2 = np.append(knot_augment, knot_augment[nk-1])
foo = np.append(((knot_aug2[(orde):]-knot_aug2[:-(orde)])/float(orde))[:-2],[0])
diff_mat = np.array([foo for i in range(len(foo)-1)])
masked_k = np.reshape(np.multiply(np.tri(N=diff_mat.shape[0], M = diff_mat.shape[1]).flatten(),diff_mat.flatten()),
                      diff_mat.shape)
aug_masked_k_1 = np.row_stack([np.zeros(len(foo)),masked_k]).T
```

```
# 3. We’ll do order 4 spline bases (i.e. cubic splines)
bs_3_1 = np.asarray(patsy.dmatrix("bs(x, knots = xi, lower_bound = 0., upper_bound = max(tau1)+0.00000000001, degree = 3,
      include_intercept=True)-1",\
             {"x": tau1}))
    # To play nicely, we need to lop off the last element
bs_4_1 = np.asarray(patsy.dmatrix("bs(x, knots = xi, lower_bound = 0., upper_bound = max(tau1)+0.00000000001, degree = 4,
      include_intercept=True)-1",\
             {"x": tau1}))[:,:-1]



#### Second group of spline stuff
## spline knots
t_obs_2 = tau2[np.array(out["delta_2"], dtype =int)==1]

xi_2= np.append(np.percentile(t_obs_2, q = np.arange(0,100,100/((knots_num)-1))),[max(tau2)])
orde = 4 ## order of the spline
## in the augmented knot matrix, you need to repeat the upper bound and lower bound orde times each
knot_augment_2 = np.append(np.append(np.repeat(0.,repeats=orde), xi_2), np.repeat(max(tau2)+0.00000000001,repeats=orde))
nk_2 = len(knot_augment_2)
knot_aug2_2 = np.append(knot_augment_2, knot_augment_2[nk_2-1])
foo_2 = np.append(((knot_aug2_2[(orde):]-knot_aug2_2[:-(orde)])/float(orde))[:-2],[0])
diff_mat_2 = np.array([foo_2 for i in range(len(foo_2)-1)])
masked_k_2 = np.reshape(np.multiply(np.tri(N=diff_mat_2.shape[0], M = diff_mat_2.shape[1]).flatten(),diff_mat_2.flatten()),
                   diff_mat_2.shape)
aug_masked_k_2 = np.row_stack([np.zeros(len(foo_2)),masked_k_2]).T
# 3. We’ll do order 4 spline bases (i.e. cubic splines)
bs_3_2 = np.asarray(patsy.dmatrix("bs(x, knots = xi_2, lower_bound = 0., upper_bound = max(tau2)+0.00000000001, degree = 3,
      include_intercept=True)-1",\
             {"x": tau2}))
    # To play nicely, we need to lop off the last element
bs_4_2 = np.asarray(patsy.dmatrix("bs(x, knots = xi_2, lower_bound = 0., upper_bound = max(tau2)+0.00000000001, degree = 4,
      include_intercept=True)-1 ",\
             {"x": tau2}))[:,:-1]
################# Models Used throughout



X_t= theano.shared(X+0.,borrow=True)
obs_t_t_1 = theano.shared(tau1+0.,borrow=True)
fail_t_1 = theano.shared(delta1+0., borrow = True)
bs_3_t_1 = theano.shared(bs_3_1+0., borrow = True)
bs_4_t_1 = theano.shared(bs_4_1+0., borrow = True)
aug_masked_k_t_1 = theano.shared(aug_masked_k_1, borrow = True)

X_t= theano.shared(X+0.,borrow=True)
obs_t_t_2 = theano.shared(tau2+0.,borrow=True)
fail_t_2 = theano.shared(delta2+0., borrow = True)
bs_3_t_2 = theano.shared(bs_3_2+0., borrow = True)
bs_4_t_2 = theano.shared(bs_4_2+0., borrow = True)
aug_masked_k_t_2 = theano.shared(aug_masked_k_2, borrow = True)
################# Models Used throughout

with pm.Model() as full_spline_clayton_penalized:
    # first instantiate the priors
    alpha = pm.HalfCauchy("alpha", 2.5)
    # first dimension
    beta_1 = pm.Normal(’beta_1’, np.zeros(2), np.ones(2)*100, shape=2)
```

```
# priors on the spline knots
# dimension of coefficients is # of knots + order, so cubic splines with 4 knots =8
gamma_0_1 = 0#pm.HalfCauchy("gamma_0_1", beta = 2.5)## intercept for penalized spline
gamma_1_1 = 0#pm.HalfCauchy("gamma_1_1", beta = 2.5)## Slope for penalized spline

# hierarchical_parameter on the spline coefficients
b_1 = pm.HalfCauchy("b_1", beta =2.5)
alpha_1 = pm.HalfCauchy(’alpha_1’,beta = b_1,shape = (1,knots_num+orde))
alpha_1.broadcastable
#### prior on the transformation parameter
r_1 = pm.InverseGamma("r_1", alpha = 1) #pm.HalfCauchy("r_1",2.5)
# second dimension
beta_2 = pm.Normal(’beta_2’, np.zeros(2), np.ones(2)*100, shape=2)
# priors on the spline knots
# dimension of coefficients is # of knots + order, so cubic splines with 4 knots = 8
gamma_0_2= 0#pm.HalfCauchy("gamma_0_2", beta = 2.5)## intercept for penalized spline
gamma_1_2 = 0#pm.HalfCauchy("gamma_1_2", beta = 2.5)## Slope for penalized spline

b_2 = pm.HalfCauchy("b_2", beta = 2.5)
alpha_2 = pm.HalfCauchy(’alpha_2’,beta = b_2,shape = (1,knots_num+orde))
alpha_2.broadcastable
#### prior on the transformation parameter
r_2 = pm.InverseGamma("r_2", alpha = 1)#pm.HalfCauchy("r_2",2.5)

## We now define the log likelihood
def logp(time_1, time_2, delta1, delta2, X):
    # Marginal Density and Survival components
    base_haz_1 = gamma_0_1+gamma_1_1*time_1+tt.dot(alpha_1, bs_3_1.T)
    base_cumhaz_1 =.5*(time_1*(2*gamma_0_1+gamma_1_1*time_1)) + tt.dot(bs_4_1,tt.dot(alpha_1,aug_masked_k_1)[0])
    base_haz_2 = gamma_0_2+gamma_1_2*time_2+tt.dot(alpha_2, bs_3_2.T)
    base_cumhaz_2 =.5*(time_2*(2*gamma_0_2+gamma_1_2*time_2)) + tt.dot(bs_4_2,tt.dot(alpha_2,aug_masked_k_2)[0])

    linear_1 = tt.dot(beta_1,X.T).T
    linear_2 = tt.dot(beta_2,X.T).T
    # next up we build parts of the likelihood
    surv_1 = tt.exp(-tt.log(1+r_1*base_cumhaz_1*tt.exp(linear_1))/r_1)
    surv_2 = tt.exp(-tt.log(1+r_2*base_cumhaz_2*tt.exp(linear_2))/r_2)
    density_1 = base_haz_1*tt.exp(linear_1)*(1+r_1*base_cumhaz_1*tt.exp(linear_1))**-(1+r_1**(-1))
    density_2 = base_haz_2*tt.exp(linear_2)*(1+r_2*base_cumhaz_2*tt.exp(linear_2))**-(1+r_2**(-1))

    # next up we build parts of the likelihood

    ### Copula derivatives:
    log_clayton_copula = (-alpha)**(-1)*tt.log(surv_1**(-alpha)+surv_2**(-alpha)-1)
    log_d_clayton_copula_s1 = -(alpha+1)*tt.log(surv_1)-((alpha+1)/alpha)*tt.log(surv_1**(-alpha)+surv_2**(-alpha)-1)
    log_d_clayton_copula_s2 = -(alpha+1)*tt.log(surv_2)-((alpha+1)/alpha)*tt.log(surv_1**(-alpha)+surv_2**(-alpha)-1)
    log_d2_clayton_copula_s1_s2 =
          tt.log(alpha+1)+(-(alpha+1))*tt.log(surv_1*surv_2)-((2*alpha+1)/alpha)*tt.log(surv_1**(-alpha)+surv_2**(-alpha)-1)
    ### different parts of log likelihood
    first = delta1*delta2*(log_d2_clayton_copula_s1_s2+tt.log(density_1)+tt.log(density_2))
    second = delta1*(1-delta2)*(log_d_clayton_copula_s1+tt.log(density_1))
    third = delta2*(1-delta1)*(log_d_clayton_copula_s2+tt.log(density_2))
    fourth = (1-delta1)*(1-delta2)*log_clayton_copula
    ### different parts of log likelihood
    return first+second+third+fourth
survival = pm.DensityDist("survival", logp,
                           observed = {"delta1": fail_t_1 , "delta2":fail_t_2,
                                       "time_1": obs_t_t_1, "time_2":obs_t_t_2,
                                       'X':X_t})
#This is what a typical call to pymc3 is...
with full_spline_clayton_penalized:
    MAP = pm.find_MAP()
    trace_HMC = pm.sample(1000,tune=n_tune, start = MAP)
```

### 5.2.2 Code for Paper 2

```
import pymc3 as pm
from pymc3 import Model, starting
import theano.tensor as tt
from theano import function as fn
import theano
import numpy as np
import scipy as sp
import pandas as pd
import random
import patsy
import argparse
### Fixed sample size for this simulation
n = 150000
## generate gamma frailty according to parameters \theta^{-1}
theta = .510
## generate gamma frailty term, ~gamma(theta^{-1},theta), np parameterizes with scale, so
w = np.random.gamma(size = n, shape=theta**(-1), scale = theta)
## functionalize this
beta1 = np.array([0.63, 0.03])
beta2 = np.array([0.8, 0.03])
beta3 = np.array([0.9, 0.03])
betas = np.vstack([beta1,beta2,beta3])
lamk = np.array([0.0047, 0.0037, 0.0057])
rhok = np.array([0.716, .725, .73])
### Simulate Datasets and Calculate MAPS
## Simulation to see what is going on here
r_this = 0.5
def sim_simple_covs(n):
    sex = np.random.binomial(n=1,p=.5,size =n)
    age = np.random.gamma(size=n, shape = 10, scale = 1/.3)
    return(np.array([sex,age]).T)
X = sim_simple_covs(n)
# Simulate Survival Times, generalize for k categories
def sim_weibull_frail_generalized(betas=betas,w=w,X1=X,lam=lamk, r = 0.0,rho=rhok,tau=365, cens_end =390,n=n):
    ## from probability integral transform
    if r == 0.0:
        r = 0.00000001
    Te = ((np.exp(-(np.log(np.random.uniform(size=(n,3)))*r)/w[:,None])-1)/(r*lam*np.exp(np.dot(X1,betas.T))))**(1/rho)
    # generate censoring time, unif(1,7) and truncated by tau
    Cens = 1+cens_end*np.random.uniform(size = n)
    Cens[Cens>tau] = tau
    alltimes = np.vstack((Cens,Te.T)).T
    eventType = []
    for i in range(len(w)):
        eventType.append(np.where(alltimes[i,]==np.amin(alltimes[i,]))[0][0])
    obs_t = list(np.amin(alltimes,axis = 1))
    out = pd.DataFrame(np.array([obs_t, eventType, pd.Series(X[:,[0]][:,0]),pd.Series(X[:,[1]][:,0]),w])).T
    out.columns = ["obs_t", "eventType", "sex", "age", "sim_frail"]
    return(out)
simulated_frail_general = sim_weibull_frail_generalized(r=r_this, X1=X, w=w)



### Instantiate the Model outside of the for loop. This reduces the computational
## define tensors with shared variables
# First, generate an instance of data to init the graph
obs_t = np.asarray(simulated_frail_general["obs_t"], dtype = float)
        #Now for bayesian Model
delta = np.asarray(pd.get_dummies(simulated_frail_general["eventType"]).drop(0.0,1), dtype = int)
        ## need this to be in nxk to play nicely with theano
tau = np.tile(np.array([obs_t]).transpose(), (1, 3))
# Design tensor
X_full = theano.shared(X+0.,borrow=True)
tau_full = theano.shared(tau+0.,borrow=True)
delta_full = theano.shared(delta+0., borrow = True)
r_t = theano.shared(0.001, borrow = True) ## placeholder value, just a dummy, will be replaced with MAP
# For the implementation of the full rank, we have a new api.
# So, we need to re-define the model
# First thing to do is to grab the Maximum a posteriori estimate
### Instantiate the Model outside of the for loop.
# This reduces the computational load
## define tensors with shared variables
# First, generate an instance of data to init the graph
############################
###### Model Specifications#
### We need to be able to find MAP overall

with pm.Model() as ordinal_surv_weibull_transformation_MAP_full:
    ## first define hyperparameter priors for weibull components
    rho = pm.Gamma(’rho’, 0.001,0.001, shape = 3)
    lam = pm.Gamma(’lam’, 0.001,0.001, shape = 3) # k dimensions
    ## define hyperparameters for gamma frailty
    theta = pm.Gamma(’theta’,0.001,0.001)
            ## create matrix of beta priors of shape kxp
    beta = pm.Normal(’beta’, np.zeros(2), np.ones(2)*100, shape = (3,2))
    ## transformation parameter
    r = pm.Gamma("r",0.01,0.01)
        ## Now define different components of log likelihood
    def logp(delta, X, tau):
        linear = tt.dot(beta,X.T).T# this is the correct formulation
        weib_base_haz = lam*rho*tau**(rho-1) #weib haz
        weib_base_cumhaz = lam*tau**(rho) # cumulative ha
        phi_1 = tt.log(weib_base_haz*np.exp(linear))
        phi_2 = tt.log((1+r*weib_base_cumhaz*np.exp(linear)))
        failed_component = tt.sum(delta*phi_1, axis = 1)-tt.sum(delta*phi_2, axis = 1)
        psi = tt.log(tt.sum(tt.log(1+r*weib_base_cumhaz*tt.exp(linear))/r,axis=1)+theta**(-1))
                    # second component for all the censored observations
        one_k = tt.ones(3)
        second = (theta**(-1)+tt.dot(delta, one_k))*psi
            # define log likelihood
        return tt.log(theta**(-tt.dot(delta,one_k))) + failed_component + theta**(-1)*tt.log(theta**(-1)) - second
    survival = pm.DensityDist('survival', logp, observed={'delta':delta_full,
                                                     'tau': tau_full,
                                                     'X': X_full})

# Run MAP on the first Model, then fill r_t tensor in computational graph
with ordinal_surv_weibull_transformation_MAP_full:
    MAP_save = pm.find_MAP()
r_MAP = np.exp(MAP_save["r_log__"])
## Now update in graph
r_t.set_value(r_MAP)

# Now define models with fixed r (the tensor)

with Model() as ordinal_surv_weibull_transformation_MAP:
    minibatch_delta = pm.Minibatch(data = delta_full.get_value(), batch_size = 3000)
    minibatch_X = pm.Minibatch(data = X_full.get_value(), batch_size = 3000)
    minibatch_tau = pm.Minibatch(data = tau_full.get_value(), batch_size = 3000)
    ## first define hyperparameter priors for weibull components
    rho = pm.Gamma('rho', 0.001,0.001, shape = 3)
    lam = pm.Gamma('lam', 0.001,0.001, shape = 3) # k dimensions
    ## define hyperparameters for gamma frailty
    theta = pm.Gamma('theta',0.001,0.001)
            ## create matrix of beta priors of shape kxp
    beta = pm.Normal('beta', np.zeros(2), np.ones(2)*100, shape = (3,2))
    ## transformation parameter
    ## This is defined as a tensor so we populate this with the MAP estimate from the overall MAP
    r = r_t
        ## Now define different components of log likelihood
    def logp(delta, X, tau):
        linear = tt.dot(beta,X.T).T# this is the correct formulation
        weib_base_haz = lam*rho*tau**(rho-1) #weib haz
        weib_base_cumhaz = lam*tau**(rho) # cumulative ha
        phi_1 = tt.log(weib_base_haz*np.exp(linear))
        phi_2 = tt.log((1+r*weib_base_cumhaz*np.exp(linear)))
        failed_component = tt.sum(delta*phi_1, axis = 1)-tt.sum(delta*phi_2, axis = 1)
        psi = tt.log(tt.sum(tt.log(1+r*weib_base_cumhaz*tt.exp(linear))/r,axis=1)+theta**(-1))
                  # second component for all the censored observations
        one_k = tt.ones(3)
        second = (theta**(-1)+tt.dot(delta, one_k))*psi
            # define log likelihood
        return tt.log(theta**(-tt.dot(delta,one_k))) + failed_component + theta**(-1)*tt.log(theta**(-1)) - second
    survival = pm.DensityDist('survival', logp, observed={'delta':delta_full,
                                                     'tau': tau_full,
                                                     'X': X_full})
with Model() as ordinal_surv_weibull_transformation_vi:
    minibatch_delta = pm.Minibatch(data = delta_full.get_value(), batch_size = 3000)
    minibatch_X = pm.Minibatch(data = X_full.get_value(), batch_size = 3000)
    minibatch_tau = pm.Minibatch(data = tau_full.get_value(), batch_size = 3000)
    ## first define hyperparameter priors for weibull components
    rho = pm.Gamma('rho', 0.001,0.001, shape = 3)
    lam = pm.Gamma('lam', 0.001,0.001, shape = 3) # k dimensions
    ## define hyperparameters for gamma frailty
    theta = pm.Gamma('theta',0.001,0.001)
            ## create matrix of beta priors of shape kxp
    beta = pm.Normal('beta', np.zeros(2), np.ones(2)*100, shape = (3,2))
    ## transformation parameter
    ## This is defined as a tensor so we populate this with the MAP estimate from the overall MAP
    r = r_t
        ## Now define different components of log likelihood
    def logp(delta, X, tau):
        linear = tt.dot(beta,X.T).T# this is the correct formulation
        weib_base_haz = lam*rho*tau**(rho-1) #weib haz
        weib_base_cumhaz = lam*tau**(rho) # cumulative ha
        phi_1 = tt.log(weib_base_haz*np.exp(linear))
        phi_2 = tt.log((1+r*weib_base_cumhaz*np.exp(linear)))
        failed_component = tt.sum(delta*phi_1, axis = 1)-tt.sum(delta*phi_2, axis = 1)
        psi = tt.log(tt.sum(tt.log(1+r*weib_base_cumhaz*tt.exp(linear))/r,axis=1)+theta**(-1))
                  # second component for all the censored observations
        one_k = tt.ones(3)
        second = (theta**(-1)+tt.dot(delta, one_k))*psi
            # define log likelihood
        return tt.log(theta**(-tt.dot(delta,one_k))) + failed_component + theta**(-1)*tt.log(theta**(-1)) - second
    survival = pm.DensityDist(’survival’, logp, observed={’delta’:minibatch_delta,
                                                    ’tau’: minibatch_tau,
                                                    ’X’: minibatch_X}, total_size = n)

# Run them

with ordinal_surv_weibull_transformation_MAP:
    MAP_save = pm.find_MAP()
with ordinal_surv_weibull_transformation_vi:
    approx2 = pm.fit(advi_iters, method = pm.FullRankADVI(), start=MAP_save)
```

# REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016), Tensorflow: a system for large-scale machine learning., *in* 'OSDI', Vol. 16, 265–283.

Ando, T. (2007), Bayesian predictive information criterion for the evaluation of hierarchical bayesian and empirical bayes models, *Biometrika* **94**(2), 443–458.

Anker, S. D. and McMurray, J. J. (2012), 'Time to move on from Ôtime-to-firstÕ: should all events be included in the analysis of clinical trials?'.

Bardenet, R., Doucet, A. and Holmes, C. (2015), On markov chain monte carlo methods for tall data, *arXiv preprint arXiv:1505.02827* .

Bastien, F., Lamblin, P., Pascanu, R., Bergstra, J., Goodfellow, I., Bergeron, A., Bouchard, N., Warde-Farley, D. and Bengio, Y. (2012), Theano: new features and speed improvements, *arXiv preprint arXiv:1211.5590* .

Beal, M. J. (2003), *Variational algorithms for approximate Bayesian inference*, University of London London.

Bedford, T. and Cooke, R. M. (2002), Vines: A new graphical model for dependent random variables, *Annals of Statistics* 1031–1068.

Bender, R., Augustin, T. and Blettner, M. (2005), Generating survival times to simulate cox proportional hazards models, *Statistics in medicine* **24**(11), 1713–1723.

Bennett, S. (1983), Analysis of survival data by the proportional odds model, *Statistics in Medicine* **2**(2), 273–277.

**URL:** *http://dx.doi.org/10.1002/sim.4780020223*

Berridge, D. M. and Whitehead, J. (1991), Analysis of failure time data with ordinal categories of response, *Statistics in medicine* **10**(11), 1703–1710.

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017), Variational inference: A review for statisticians, *Journal of the American Statistical Association* (just-accepted).

Brooks, S., Gelman, A., Jones, G. and Meng, X.-L. (2011), *Handbook of markov chain monte carlo*, CRC press.

Carroll, K. N., Gebretsadik, T., Griffin, M. R., Wu, P., Dupont, W. D., Mitchel, E. F., Enriquez, R. and Hartert, T. V. (2008), Increasing burden and risk factors for bronchiolitis-related medical visits in infants enrolled in a state health care insurance plan, *Pediatrics* **122**(1), 58–64.

Carroll, K. N., Wu, P., Gebretsadik, T., Griffin, M. R., Dupont, W. D., Mitchel, E. F. and Hartert, T. V. (2009), The severity-dependent relationship of infant bronchiolitis on the risk and morbidity of early childhood asthma, *Journal of Allergy and Clinical Immunology* **123**(5), 1055–1061.

Casella, G. and Berger, R. L. (2002), *Statistical inference*, Vol. 2, Duxbury Pacific Grove, CA.

Cave, D. G. and Munson, B. (1999), 'Medical claims integration and data analysis system'. US Patent 5,970,463.

Chen, C.-M. and Lu, T.-F. C. (2012), Marginal analysis of multivariate failure time data with a surviving fraction based on semiparametric transformation cure models, *Computational Statistics  Data Analysis* **56**(3), 645 – 655.
**URL:** *http://www.sciencedirect.com/science/article/pii/S016794731100332X*

Chen, C.-M. and Yu, C.-Y. (2012), A two-stage estimation in the clayton–oakes model with marginal linear transformation models for multivariate failure time data, *Lifetime data analysis* **18**(1), 94–115.

Chen, Y.-H. (2010), Semiparametric marginal regression analysis for dependent competing risks under an assumed copula, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **72**(2).

Chib, S. and Greenberg, E. (1995), Understanding the metropolis-hastings algorithm, *The american statistician* **49**(4), 327–335.

Claggett, B., Pocock, S., Wei, L., Pfeffer, M. A., McMurray, J. J. and Solomon, S. D. (2018), Comparison of time-to-first event and recurrent-event methods in randomized clinical trials, *Circulation* **138**(6), 570–577.

Clayton, D. and Cuzick, J. (1985), Multivariate generalizations of the proportional hazards model, *Journal of the Royal Statistical Society. Series A (General)* 82–117.

Craiu, V. R. and Sabeti, A. (2012), In mixed company: Bayesian inference for bivariate conditional copula models with discrete and continuous outcomes, *Journal of Multivariate Analysis* **110**, 106 – 120. Special Issue on Copula Modeling and Dependence.

**URL:** *http://www.sciencedirect.com/science/article/pii/S0047259X12000796*

Dabrowska, D. M. and Doksum, K. A. (1988), Estimation and testing in a two-sample generalized odds-rate model, *Journal of the American Statistical Association* **83**(403), 744–749.

Davis, P. J. and Rabinowitz, P. (2007), *Methods of numerical integration*, Courier Corporation.

De Boor, C., De Boor, C., Mathématicien, E.-U., De Boor, C. and De Boor, C. (1978), *A practical guide to splines*, Vol. 27, Springer-Verlag New York.

de Castro, M., Chen, M.-H., Ibrahim, J. G. and Klein, J. P. (2014), Bayesian transformation models for multivariate survival data, *Scandinavian Journal of Statistics* **41**(1), 187–199.
**URL:** *http://dx.doi.org/10.1111/sjos.12010*

Diao, G. and Yin, G. (2012), A general transformation class of semiparametric cure rate frailty models, *Annals of the Institute of Statistical Mathematics* **64**(5), 959–989.
**URL:** *https://doi.org/10.1007/s10463-012-0354-0*

E Shemyakin, A. and Youn, H. (2006), Copula models of joint last survivor analysis, *Applied Stochastic Models in Business and Industry* **22**(2), 211–224.

Efron, B. (1992), Bootstrap methods: another look at the jackknife, *in* 'Breakthroughs in statistics', Springer, 569–593.

Falcaro, M. and Pickles, A. (2007), A flexible model for multivariate interval-censored survival times with complex correlation structure, *Statistics in medicine* **26**(3), 663–680.

Fernández, T., Rivera, N. and Teh, Y. W. (2016), Gaussian processes for survival analysis, *in* 'Advances in Neural Information Processing Systems', 5021–5029.

Geisser, S., Hodges, J., Press, S. and ZeUner, A. (1990), The validity of posterior expansions based on laplaceâĂŹs method, *Bayesian and likelihood methods in statistics and econometrics* **7**, 473.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2014), *Bayesian data analysis*, Vol. 2, CRC press Boca Raton, FL.

Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013), *Bayesian data analysis*, Chapman and Hall/CRC.

Genest, C. and Mackay, J. (1986), The joy of copulas: Bivariate distributions with uniform marginals, *The American Statistician* **40**(4), 280–283.
**URL:** *https://www.tandfonline.com/doi/abs/10.1080/00031305.1986.10475414*

Georges, P., Lamy, A.-G., Nicolas, E., Quibel, G. and Roncalli, T. (2001), Multivariate survival modelling: a unified approach with copulas.

Goethals, K., Janssen, P. and Duchateau, L. (2008), Frailty models and copulas: similarities and differences, *Journal of Applied Statistics* **35**(9), 1071–1079.

Graves, A. (2011), Practical variational inference for neural networks, *in* 'Advances in neural information processing systems', 2348–2356.

Hartert, T. V., Carroll, K., Gebretsadik, T., Woodward, K. and Minton, P. (2010), The tennessee children's respiratory initiative: Objectives, design and recruitment results of a prospective cohort study investigating infant viral respiratory illness and the development of asthma and allergic diseases, *Respirology* **15**(4), 691–699.

Hedeker, D., Siddiqui, O. and Hu, F. B. (2000), Random-effects regression analysis of correlated grouped-time survival data, *Statistical Methods in Medical Research* **9**(2), 161–179.

Herndon, J. E. and Harrell Jr, F. E. (1995), The restricted cubic spline as baseline hazard in the proportional hazards model with step function time-dependent covariables, *Statistics in medicine* **14**(19), 2119–2129.

Hoffman, M. D., Blei, D. M., Wang, C. and Paisley, J. (2013), Stochastic variational inference, *The Journal of Machine Learning Research* **14**(1), 1303–1347.

Hoffman, M. D. and Gelman, A. (2014), The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo., *Journal of Machine Learning Research* **15**(1), 1593–1623.

Ibrahim, J. G., Chen, M.-H. and Sinha, D. (2005), *Bayesian survival analysis*, Wiley Online Library.

Joe, H. (1997), *Multivariate models and multivariate dependence concepts*, CRC Press.

Klein, J. P. and Moeschberger, M. L. (2005), *Survival analysis: techniques for censored and truncated data*, Springer Science & Business Media.

Kucukelbir, A., Ranganath, R., Gelman, A. and Blei, D. M. (2015), 'Automatic variational inference in stan'.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J. and Sohl-Dickstein, J. (2017), Deep neural networks as gaussian processes, *arXiv preprint arXiv:1711.00165* .

Li, R., Cheng, Y., Chen, Q. and Fine, J. (2017), Quantile association for bivariate survival data, *Biometrics* **73**(2), 506–516.

Li, Y., Prentice, R. L. and Lin, X. (2008), Semiparametric maximum likelihood estimation in normal transformation models for bivariate survival data, *Biometrika* **95**(4), 947–960.

Lin, H., Zhou, L., Li, C. and Li, Y. (2014), Semiparametric transformation models for semicompeting survival data, *Biometrics* **70**(3), 599–607.

Lin, Y., Luo, Y., Xie, S. and Chen, K. (2017), Robust rank estimation for transformation models with random effects, *Biometrika* **104**(4), 971–986.
**URL:** *http://dx.doi.org/10.1093/biomet/asx055*

Louzada, F., Suzuki, A. and Cancho, V. (2013), The fgm long-term bivariate survival copula model: modeling, bayesian estimation, and case influence diagnostics, *Communications in Statistics-Theory and Methods* **42**(4), 673–691.

Marshall, A. W. and Olkin, I. (1988), Families of multivariate distributions, *Journal of the American statistical association* **83**(403), 834–841.

Masarotto, G., Varin, C. et al. (2012), Gaussian copula marginal regression, *Electronic Journal of Statistics* **6**, 1517–1549.

McCallum, G. B., Morris, P. S., Wilson, C. C., Versteegh, L. A., Ward, L. M., Chatfield, M. D. and Chang, A. B. (2013), Severity scoring systems: are they internally valid, reliable and predictive of oxygen use in children with acute bronchiolitis?, *Pediatric pulmonology* **48**(8), 797–803.

Meissner, H. C. (2016), Viral bronchiolitis in children, *New England Journal of Medicine* **374**(1), 62–72.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953), Equation of state calculations by fast computing machines, *The journal of chemical physics* **21**(6), 1087–1092.

Meyer, R. and Romeo, J. S. (2015), Bayesian semiparametric analysis of recurrent failure time data using copulas, *Biometrical Journal* **57**(6), 982–1001.

Minka, T. P. (2001), Expectation propagation for approximate bayesian inference, *in* ‘Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence’, Morgan Kaufmann Publishers Inc., 362–369.

Nadarajah, S., Afuecheta, E. and Chan, S. (2017), A compendium of copulas, *Statistica* **77**(4), 279.

Nelsen, R. B. (1999), ‘An introduction to copulas, volume 139 of lecture notes in statistics’.

Nelsen, R. B. (2007), *An introduction to copulas*, Springer Science & Business Media.

Oakes, D. (1989), Bivariate survival models induced by frailties, *Journal of the American Statistical Association* **84**(406), 487–493.

Python Core Team (2015), *Python: A dynamic, open source programming language*, Python Software Foundation.
**URL:** *https://www.python.org/*

R Core Team (2014), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
**URL:** *http://www.R-project.org/*

Rasmussen, C. E. (2004), Gaussian processes in machine learning, *in* ‘Advanced lectures on machine learning’, Springer, 63–71.

Rasmussen, C. E. and Williams, C. K. (2006), *Gaussian processes for machine learning*, Vol. 1, MIT press Cambridge.

Rauch, G., Kieser, M., Binder, H., Bayes-Genis, A. and Jahn-Eimermacher, A. (2018), Time-to-first-event versus recurrent-event analysis: points to consider for selecting a meaningful analysis strategy in clinical trials with composite endpoints, *Clinical Research in Cardiology* **107**(5), 437–443.

Rezende, D. J. and Mohamed, S. (2015), Variational inference with normalizing flows, *arXiv preprint arXiv:1505.05770* .

Rodriguez, H., Hartert, T. V., Gebretsadik, T., Carroll, K. N. and Larkin, E. K. (2016), A simple respiratory severity score that may be used in evaluation of acute respiratory infection, *BMC research notes* **9**(1), 85.

Romeo, J. S., Meyer, R. and Gallardo, D. I. (2018), Bayesian bivariate survival analysis using the power variance function copula, *Lifetime data analysis* **24**(2), 355–383.

Romeo, J. S., Tanaka, N. I. and Pedroso-de Lima, A. C. (2006), Bivariate survival modeling: a bayesian approach based on copulas, *Lifetime Data Analysis* **12**(2), 205–222.

Salvatier J, Wiecki TV, F. C. (2016), 'Probabilistic programming in python using pymc3'.
**URL:** *https://doi.org/10.7717/peerj-cs.55*

Sargent, D. J. (1998), A general framework for random effects survival analysis in the cox proportional hazards setting, *Biometrics* 1486–1497.

Schmidt, M. I., Duncan, B. B., Sharrett, A. R., Lindberg, G., Savage, P. J., Offenbacher, S., Azambuja, M. I., Tracy, R. P., Heiss, G., investigators, A. et al. (1999), Markers of inflammation and prediction of diabetes mellitus in adults (atherosclerosis risk in communities study): a cohort study, *The Lancet* **353**(9165), 1649–1652.

Sharef, E., Strawderman, R. L., Ruppert, D., Cowen, M., Halasyamani, L. et al. (2010), Bayesian adaptive b-spline estimation in proportional hazards frailty models, *Electronic journal of statistics* **4**, 606–642.

Shih, J. H. and Louis, T. A. (1995), Inferences on the association parameter in copula models for bivariate survival data, *Biometrics* 1384–1399.

Shun, Z. and McCullagh, P. (1995), Laplace approximation of high dimensional integrals, *Journal of the Royal Statistical Society. Series B (Methodological)* 749–760.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P. and Van Der Linde, A. (2002), Bayesian measures of model complexity and fit, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **64**(4), 583–639.

Spiegelhalter, D., Thomas, A., Best, N., Gilks, W. and Lunn, D. (1994), Bugs: Bayesian inference using gibbs sampling. mrc biostatistics unit, cambridge, england, *URL: http://www. mrc-bsu. cam. ac. uk/bugs* **21**, 27.

Spiegelhalter, D., Thomas, A., Best, N. and Lunn, D. (2007), Openbugs user manual, version 3.0. 2, *MRC Biostatistics Unit, Cambridge* .

Srivastava, S., Cevher, V., Dinh, Q. and Dunson, D. (2015), Wasp: Scalable bayes via barycenters of subset posteriors, *in* 'Artificial Intelligence and Statistics', 912–920.

Sueyoshi, G. T. (1992), Semiparametric proportional hazards estimation of competing risks models with time-varying covariates, *Journal of econometrics* **51**(1-2), 25–58.

Theano Development Team (2016), Theano: A Python framework for fast computation of mathematical expressions, *arXiv e-prints* **abs/1605.02688**.
**URL:** *http://arxiv.org/abs/1605.02688*

Thomas, R. and Have, T. (1996), A mixed effects model for multivariate ordinal response data including correlated discrete failure times with ordinal responses, *Biometrics* 473–491.

Vehtari, A., Gelman, A. and Gabry, J. (2017), Practical bayesian model evaluation using leave-one-out cross-validation and waic, *Statistics and Computing* **27**(5), 1413–1432.

Wang, B. and Titterington, D. (2012), Convergence and asymptotic normality of variational bayesian approximations for exponential family models with missing values, *arXiv preprint arXiv:1207.4159* .

Wei, L.-J. (1992), The accelerated failure time model: a useful alternative to the cox regression model in survival analysis, *Statistics in medicine* **11**(14-15), 1871–1879.

Widder, D. V. (2015), *Laplace transform (PMS-6)*, Princeton university press.

Wu, P., Dupont, W. D., Griffin, M. R., Carroll, K. N., Mitchel, E. F., Gebretsadik, T. and Hartert, T. V. (2008), Evidence of a causal role of winter virus infection during infancy in early childhood asthma, *American journal of respiratory and critical care medicine* **178**(11), 1123–1129.

Yin, G. (2008), Bayesian transformation cure frailty models with multivariate failure time data, *Statistics in medicine* **27**(28), 5929–5940.

Zeng, D., Chen, Q. and Ibrahim, J. G. (2009), Gamma frailty transformation models for multivariate survival times, *Biometrika* **96**(2), 277–291.

Zeng, D. and Lin, D. (2007), Maximum likelihood estimation in semiparametric regression models with censored data, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**(4), 507–564.