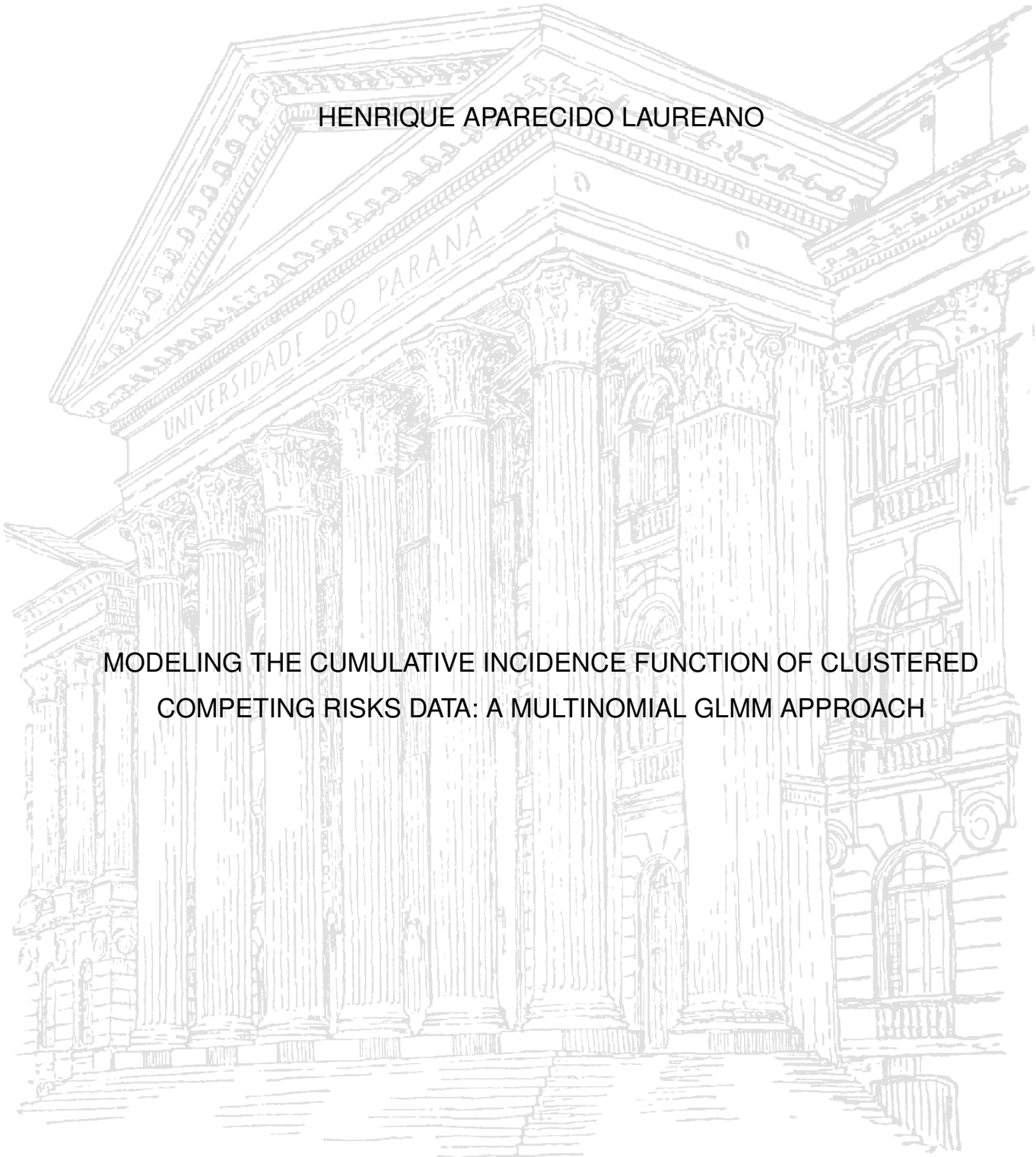
FEDERAL UNIVERSITY OF PARANÁ

HENRIQUE APARECIDO LAUREANO



MODELING THE CUMULATIVE INCIDENCE FUNCTION OF CLUSTERED COMPETING RISKS DATA: A MULTINOMIAL GLMM APPROACH

CURITIBA

2021

HENRIQUE APARECIDO LAUREANO

MODELING THE CUMULATIVE INCIDENCE FUNCTION OF CLUSTERED COMPETING RISKS DATA: A MULTINOMIAL GLMM APPROACH

Thesis presented to the Graduate Program of Numerical Methods in Engineering, Concentration Area in Mathematical Programming: Statistical Methods Applied in Engineering, Federal University of Paraná, as part of the requirements to the obtention of the Master's Degree in Sciences.

Supervisor: Prof. PhD Wagner Hugo Bonat

Co-supervisor: Prof. PhD Paulo Justiniano Ribeiro Jr

CURITIBA

2021

HENRIQUE APARECIDO LAUREANO

MODELING THE CUMULATIVE INCIDENCE FUNCTION OF CLUSTERED COMPETING RISKS DATA: A MULTINOMIAL GLMM APPROACH

Thesis presented to the Graduate Program of Numerical Methods in Engineering, Concentration Area in Mathematical Programming: Statistical Methods Applied in Engineering, Federal University of Paraná, as part of the requirements to the obtention of the Master's Degree in Sciences.

Master thesis approved. XXX XX, 2021.

Prof. PhD Wagner Hugo Bonat
Supervisor

Prof. PhD Paulo Justiniano Ribeiro Jr
Co-supervisor

Prof. PhD ...
Internal Examinator - PPGMNE

Prof. PhD ...
Internal Examinator - PPGMNE

Prof. PhD ...
External Examiner -

CURITIBA
2021

To Celita and Olivio

ACKNOWLEDGEMENTS

As Moro said once, I'm thankful for everything and everyone.

"It's not supposed to be easy."
(Gregg Popovich)

ABSTRACT

Failure time data …

Keywords: Competing risks.

RESUMO

Dados de tempos de falha …

Palavras-chave: Riscos competitivos.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CONTENTS

1 INTRODUCTION

Consider a cluster of random variables representing the time until the occurrence of some event. These random variables are assumed to be correlated, i.e. for some biological or environmental reason it is not adequate to assume independence between them. Also, we may be interested in the occurrence of not only one specific event, having in practice a competition of events to see which one happens first, if it happens. Such events may also be of low probability albeit severe consequences, this is the moment when the cluster correlation makes its difference: the occurrence of an event in a cluster member should affect the probability of the same happening in the others.

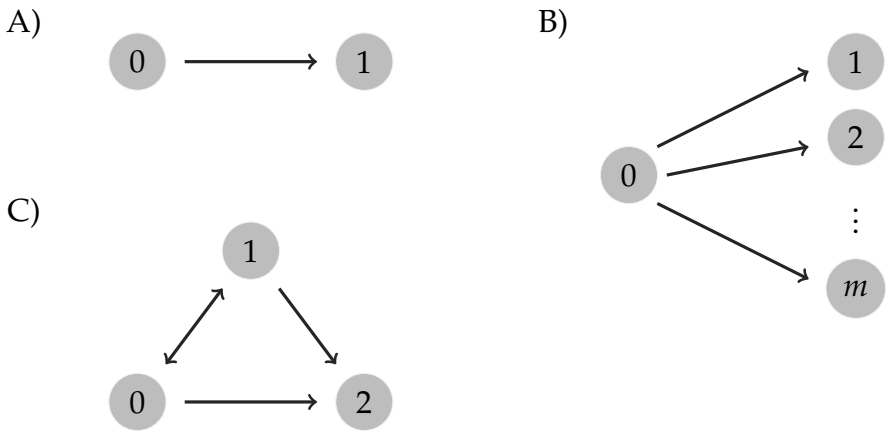
A realistic context that fits perfectly with the framework described above is the study of disease incidence in family members, where each member is indexed by a random variable and each cluster consists of a familiar structure. The inspiration to the study of these kinds of problems came from the work developed in Cederkvist et al. (2019), where they studied breast cancer incidence in mothers and daughters but using a complicated modeling framework. Based on that, the aim of this thesis is to propose a simpler framework taking advantage of several *state-of-art* computational libraries and see how far can we go in several scenarios. Until now we just contextualized, we still need to introduce the methodology. To this, some definitions and theoretical contexts are welcome.

When the object under study is a random variable representing the time until some event occurs, we are in the field of *failure time data* (KALBFLEISCH; PRENTICE, 2002). The occurrence of an event is generally denoted *failure*, and major areas of application are biomedical studies and industrial life testing. In this thesis, we maintain our focus on the former. As common in science, same methodologies can receive different names depending on the area. In industrial life testing is performed what is called a *reliability analysis*; in biomedical studies is performed what is called *survival analysis*. Generally, the term *survival* is applied when we are interested in the occurrence of only one event, a *failure time process*. When we are interested in the occurrence of more than one event we enter in the yard of *competing risks* and *multistate* models. A visual aid is presented on Figure 1 and a comprehensive reference is Kalbfleisch & Prentice (2002).

Failure time and competing risks processes may be seen as particular cases of a multistate model. Besides the number of events (states) of interest, the main difference between a multistate model and its particular cases is that only in the multistate scenario we may have transient states, using a *stochastic process* language. In the particular cases, all states besides the initial state 0, are absorbents - once you reached it you do not leave.

The simplest multistate model that exemplify this behavior is the illness-death model, Figure 1 C), where a patient (initially in state 0) can get sick (state 1) or die (state 2); if sick it can recover (returns to state 0) or die. We work in this thesis only with competing risks processes, and to each patient we need the time (age) until the occurrence, or not, of the event.

FIGURE 1 – ILLUSTRATION OF MULTISTATE MODELS FOR A A) FAILURE TIME PROCESS; B) COMPETING RISKS PROCESS; AND C) ILNESS-DEATH MODEL, THE SIMPLEST MULTISTATE MODEL



SOURCE: The author (2021).

When for some know or unknown reason we are not able to see the occurrence of an event, we have what is denoted *censorship*. Still in the illness-death model, during the period of follow up the patient may not get sick or die, staying at state 0. This is denoted *right-censorship*; If a patient is in state 1 at the end of the study, we are *censored* to see him reaching the state 2 or returning to state 0. This is the inherent idea to censorship and must be present in the modeling framework, thus arriving in the so-called *survival models* (KALBFLEISCH; PRENTICE, 2002).

A survival model deals with the survival experience. Usually, the survival experience is modeled in the *hazard* (failure rate) scale and it can be expressed for a subject i as

$$\lambda(t \mid \boldsymbol{x}_i) = \lambda_0(t) \times c(\boldsymbol{x}_i\boldsymbol{\beta}) \quad \text{at time } t, \tag{1.1}$$

i.e. as the product of an arbitrary baseline hazard function $\lambda_0(\cdot)$, with a specific function form $c(\cdot)$, that will depend on the probability distribution to be chosen for the failure time and on predictors/covariates/explanatory/independent variables $\boldsymbol{x}_i = [x_1 \ \ldots \ x_p]$, where $\boldsymbol{\beta}^\top = [\beta_1 \ \ldots \ \beta_p]$ is the parameters vector.

This structure is specified for a failure time process, as in Figure 1 A). Nevertheless, the idea is easy to extend. We basically have the Equation 1.1's model to each cause-specific (in a competing risks process) or transition (in a multistate process). A complete and extensive detailing can be, again, found in Kalbfleisch & Prentice (2002).

In this work we approach the case of clustered competing risks. Besides the cause-specific structure, we have to deal with the fact that the events are happening in related individuals. This configures what is denoted *family studies*, i.e. we have a cluster/group/family dependence that needs to be considered, accommodated, and modeled. This, possible, dependence is something that we do not actually measure but know (or just suppose) that exists. In the statistical modeling language this characteristic receives the name of *random* or *latent effect*. A survival model with a latent effect, association, or unobserved heterogeneity, is denoted *frailty model* (CLAYTON, 1978; VALPEL; MANTON; STALLARD, 1979). In its simplest form, a frailty is an unobserved random proportionality factor that modifies the hazard function of an individual, or of related individuals. Frailty models are extensions of Equation 1.1's model.

In the competing risks setting, the hazard scale (focusing on the cause-specific hazard) is not the only possible scale to work on. A more attractive possibility is to work on the probability scale (ANDERSEN et al., 2012), focusing on the cause-specific cumulative incidence function (CIF). Besides the within-family dependence, in family studies there is often a strong interest in describing age at disease onset, which is directly described by the cause-specific CIF. Therefore, making the probability scale a more attractive and logical choice. Since the CIF plays a central role in this master thesis, it will be formally defined later in a place with greater emphasis. With the definitions and the theoretical context being made, let us be more specific.

To work with competing risks data on the probability scale plus a latent structure allowing for within-cluster dependence of both risk and timing, Cederkvist et al. (2019) proposed a pairwise composite likelihood approach based on the factorization of the cause-specific CIF as the product of a cluster-specific risk level function with a cluster-specific failure time trajectory function. A composite approach (LINDSAY, 1988; COX; REID, 2004; VARIN; REID; FIRTH, 2011) is a valid alternative to a full likelihood analysis in high-dimensional situations when a full approach is too computational costly or even inviable. In failure time data problems, the composite likelihood function is built from the product of marginal densities. The marginal specification implies a pairwise approach since we need to add model layers to be able to handle with the dependence structure. A clear advantage of this approach is that we do not need to care about a joint distribution specification, which generally translates also into a computational advantage. A disadvantage is the model specification, which becomes much more complicated, besides the number of small details to be workaround from the fact of being working with not an exact likelihood function.

We do not have any guarantees that a full likelihood inference procedure is not viable here, so we try to reach the same goal of Cederkvist et al. (2019) albeit with a simpler framework taking advantage of *state-of-art* software, something still

not so common in the statistical modeling community. This simpler framework is a generalized linear mixed model (GLMM). Instead of concentrating on failure time data and consequently having a survival/frailty model based on the hazard scale, or using a composite approach, we just build the joint/full likelihood function (a multinomial model with its link function based on the cluster-specific CIF, accouting for an appropriate latent effects structure), marginalize (integrate out the latent effects) and optimize it. A Fisherian approach per se.

To a better contextualization of our GLMM approach (MCCULLOCH; SEARLE, 2001), consider a random suject i. In a standard linear model we assume that the response variable Y_i, conditioned on the covariates $\boldsymbol{x}_i$, follows a normal/Gaussian distribution and what we do is to model its mean, $\mu_i \equiv \mathbb{E}(Y_i \mid \boldsymbol{x}_i)$, via a linear combination. As much well explained in Nelder & Wedderburn (1972), with the aid of a *link function* $g(\cdot)$, this idea is generalized to distributions of the *exponential family*. Many of its members are useful for practical modelling, such as the Poisson (for counting data), binomial (dichotomic data), gamma (continuous but positive) and Gaussian (continuous data) distributions. This extended framework received the name of generalized linear model (GLM) and a comprehensive reference is McCullagh & Nelder (1989).

What makes a GLM into a GLMM (MCCULLOCH; SEARLE, 2001) is the addition of a latent effect $\boldsymbol{u}$ (then, *m*ixed) into the mean structure. The mean structure of a standard GLMM is defined as

$$g(\mu_i) = \boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{u}, \quad \boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$$

where the latent effect is assumed to follow a multivariate Gaussian distribution of zero mean and a parametrized variance-covariance matrix $\boldsymbol{\Sigma}$. Its correct linkage to the mean structure is made through the i^{th} vector row of a design-matrix $\boldsymbol{Z}$. The covariates are into $\boldsymbol{x}_i$, the i^{th} vector row of a model-matrix $\boldsymbol{X}$, with $\boldsymbol{\beta}$ being a vector of unknown parameters.

1.1 GOALS

1.1.1 General goals

Propose and study the estimability of a multinomial generalized linear mixed model (multiGLMM) to the cluster and cause-specific cumulative incidence function (CIF) of clustered competing risks data.

1.1.2 Specific goals

1. Simulate from the model, i.e. generate synthetic data to study statistical properties.

2. Write the model in the Template Model Builder (TMB) software, developed by Kristensen et al. (2016) and possibly the most efficient likelihood-based way of doing such task.

3. Take advantage of TMB's functionalities with special attention to the computation of gradients and Hessians via a *state-of-art* automatic differentiation (AD) implementation; and a joint likelihood marginalization via an efficient Laplace approximation routine.

4. Study the model identifiability through the proposition of different complexity level models in terms of parametric space and latent effect structures.

5. Make exact likelihood-based inference to the cluster and cause-specific CIF of clustered competing risks data.

1.2 JUSTIFICATION

In the biomedical statistical modeling literature, the study of disease occurrence in related individuals receives the name of family studies. Key points of interest are the within-family dependence and determining the role of different risk factors. The within-family dependence may reflect both disease heritability and the impact of shared environmental effects. The role of different risk factors arrives in the class of multivariate models, which options are limited in the statistical literature. Thus, the number of statistical models for competing risks data that accommodate the within-cluster/family dependence is even more limited. Some modeling options are briefly commented in Cederkvist et al. (2019), with his pairwise composite approach being proposed as a new and better option to model the cause-specific cumulative incidence function (CIF), describing age at disease onset, of clustered competing risks data on the probability scale. We propose to model the cause-specific CIF and accommodate the within-family dependence in the same fashion (via a latent structure that allows the absolute risk and the failure time distribution to vary between families) but with an easier framework, based on a multinomial generalized linear mixed model approach.

1.3 LIMITATION

This work restraint to the proposition and model identifiability study of a multinomial model for the cause-specific cumulative incidence function (CIF) of competing risks data, with a latent effect structure to accommodate within-family dependence with regard to both risk and timing. Given its considerable model complexity, hypothesis tests; residual analysis; and good-of-fit measures are not contemplated.

1.4 THESIS ORGANIZATION

This master thesis contains 6 chapters including this introduction. Chapter 2 presents a systematic review of the main aspects involved in the formulation and optimzation of a generalized linear mixed model (GLMM). Given the modeling framework overview, Chapter 3 presents our multinomial GLMM (multiGLMM) to model the cause-specific cumulative incidence function (CIF) of clustered competing risks data. In Chapter 4 we describes the simulation procedure to generate synthetic data and present some model particularities. In Chapter 5 the obtained results are presented, and in Chapter 6 we discuss the contributions of this thesis and present some suggestions for future work.

2 GENERALIZED LINEAR MIXED MODELS: FORMULATION, OPTIMIZATION, AND IMPLEMENTATION

This chapter presents a systematic review of the main theoretical aspects involved in the formulation, estimation, and implementation of a generalized linear mixed model (GLMM). We start in Section 2.1 with the model formulation framework, concluding with the so-called joint or full likelihood function. Section 2.2 address the marginalization of that joint likelihood, performed here in terms of a Laplace approximation technique. Section 2.3 discusses available alternatives for the marginal likelihood parameters optimization. Section 2.4 present the automatic differentiation (AD) procedure, the most efficent routine for the computation of derivatives, and a key point for us. Last but not least, in Section 2.5 we present the computational tool used to peform all the discussed methodology, a very exciting R (R Core Team, 2021) package called TMB: Template Model Builder, developed by Kristensen et al. (2016).

2.1 FORMULATION: OBTAINING A JOINT LIKELIHOOD FUNCTION

We model an n-vector of exponential family random variables $\boldsymbol{Y}$, in terms of its conditional expected value $\boldsymbol{\mu} \equiv \mathbb{E}(\boldsymbol{Y} \mid \boldsymbol{X}, \boldsymbol{u})$, via a linear combination called of linear predictor and generally expressed by

$$g(\boldsymbol{\mu}) = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u}, \quad \boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}). \tag{2.1}$$

In other words, a GLMM (MCCULLOCH; SEARLE, 2001) is a generalized linear model (GLM) in which the linear predictor depends on some Gaussian latent effects, $\boldsymbol{u}$, times a latent effects design-matrix $\boldsymbol{Z}$. Since we do not observe the latent component, an exemplification of the idea embedded in matrix $\boldsymbol{Z}$ is welcome. Suppose e.g., three individuals (or clusters) and that each one has two measures. This configures a repeated measures context, the most common latent structure in family studies. Also, it is reasonable to admit that each individual has its particular latent effect value. Consequently, we have

$$\boldsymbol{Z}\boldsymbol{u} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_1 \\ u_2 \\ u_2 \\ u_3 \\ u_3 \end{bmatrix},$$

where $\boldsymbol{u}^\top = [u_1 \; u_2 \; u_3]$ and $\boldsymbol{Z}$ has the role of projecting the values of $\boldsymbol{u}$ to match the number of measures.

In a mixed model the mean structure is approached as a combination of probability distributions. It is a combination since we have to assume probabilistic structures for the observed and non-observed/latent data. To each observed variable y_{ij} we have a probability distribution of the exponential family, denoted by $f(y_{ij} \mid \boldsymbol{u}_i, \boldsymbol{\theta})$. To the latent effect we have, generally, a (multivariate) Gaussian distribution, denoted by $f(\boldsymbol{u}_i \mid \boldsymbol{\Sigma})$. To each individual or unity under study i, and to each measure j, we have the product of these probability densities, a likelihood contribution.

Our goal is to estimate the parameter vector $\boldsymbol{\theta} = [\boldsymbol{\beta}\ \boldsymbol{\Sigma}]^\top$ of a mean structure, as in Equation 2.1. Besides the role of emphasizing the fact that $\boldsymbol{\mu}$ is a function of $\boldsymbol{\theta}$, and that we want to estimate $\boldsymbol{\theta}$, the likelihood function ties the probability densities i.e., the likelihood is the product of the product of probability densities, to each subject i. Since Y_i are mutually independent, the likelihood for $\boldsymbol{\theta}$ can be written as

$$L(\boldsymbol{\theta} \mid \boldsymbol{y}, \boldsymbol{u}) = \prod_{i=1}^{I} \prod_{j=1}^{n_i} f(y_{ij} \mid \boldsymbol{u}_i, \boldsymbol{\beta}, \boldsymbol{\Sigma})\ f(\boldsymbol{u}_i \mid \boldsymbol{\Sigma}). \tag{2.2}$$

From standard probability theory is easy to see that in the right-hand side (r.h.s.) we have a joint density, consequently, Equation 2.2 represents what is called a full or a joint likelihood function. What makes problematic working with this joint likelihood is that we do not have all the necessary information to just maximize it and get the desired parameter estimates. The latent effect $\boldsymbol{u}$ is *latent* i.e., we do not observe it. To handle this we have basically two available paths.

2.2 MARGINALIZATION: LAPLACE APPROXIMATION AND ALTERNATIVES

To deal with a joint likelihood function as in Equation 2.2 we have a choice to make. Be or not to be Bayesian. Each choice has its own difficulties, advantages, and characteristics.

The Bayesian path assumes that all $\boldsymbol{\theta}$ components are random variables. With all parameters being treated as random variables, and since we do not observe them, what the Bayesian framework does is try to compute the mode of each "parameter" marginal distribution, generally, via a sampling algorithm called MCMC: Markov chain Monte Carlo (GELFAND; SMITH, 1990; DIACONIS, 2009).

The advantage of being Bayesian is that we can reach an MCMC algorithm to basically any statistical model, the disadvantage is that this approach is very time consuming and we have to propose prior distributions to each "parameter". These prior proposals are not always easy to make, and the resulting marginal distributions can be very depending of it. A Bayesian approach can be applied in basically any context, without guarantees that will work - obtain convergence to all parameters is not

a straightforward task. However, in complex scenarios they can be the only available method to "maximize" the likelihood function. This is not the case here.

We have a joint density where one of the random variables is not observed, but we are not interested in it, only in the variance parameters inherent in it. Again, from standard probability theory, if we have a joint density we can just integrate out the undesired variable resulting in

$$\begin{aligned} L(\boldsymbol{\theta} \mid \boldsymbol{y}) &= \prod_{i=1}^{I} \int_{\mathcal{R}^{u_i}} \left[\prod_{j=1}^{n_i} f(y_{ij} \mid \boldsymbol{u}_i, \boldsymbol{\beta}, \boldsymbol{\Sigma}) \, f(\boldsymbol{u}_i \mid \boldsymbol{\Sigma}) \right] \mathrm{d}\boldsymbol{u}_i \\ &= \prod_{i=1}^{I} \int_{\mathcal{R}^{u_i}} f(\boldsymbol{y}_i, \boldsymbol{u}_i \mid \boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{u}_i, \end{aligned} \tag{2.3}$$

a marginal density that keeps the parameters $\boldsymbol{\Sigma}$ of the integrated variable.

When the response distribution of a mixed model is Gaussian, is analytically tractable to integrate $\boldsymbol{u}$ out of the joint density. Consequently, it is possible to evaluate the marginal likelihood exactly. This is the case of the linear mixed models (LMMs) and one of the main differences to the GLMMs. When the response distribution is not Gaussian, generally, it is not anymore analytically tractable to integrate out the latent effect. So what do we do? Well, we have basically two options again.

We can avoid the integrals in Equation 2.3, replacing it by integrals that are more analytically tractable. This can be performed via an algorithm called Expectation-Maximization (EM), proposed by Dempster, Laird & Rubin (1977). This approach is considered a little bit naive and generally is not recommended if you have a better option. The other option consists of performing a numerical integration i.e., approximating the integral. The most common way of doing that in the statistical modeling literature is via an importance sampling version of the Gaussian quadrature rule, denoted adaptive Gaussian quadrature (AGQ) (PINHEIRO; CHAO, 2006). In general, adaptive Gaussian quadratures are not so simple to use (computationally expensive; we have to choose how many integration points will be used; and we also have to choose an importance distribution to approximate the integrand).

To us, the better option consists in take advantage of the exponential family structure together with the fact that we are dealing with Gaussian latent effects. These ideas converge to an adaptive Gaussian quadrature with one integration point, also called as *Laplace approximation* (MOLENBERGHS; VERBEKE, 2005; SHUN; MCCULLAGH, 1995; TIERNEY; KADANE, 1986; WOOD, 2015).

With an integral that is analytically intractable, we may approximate it to obtain a tractable closed-form expression allowing then the numerical maximization of the resulting marginal likelihood function (BONAT; RIBEIRO, 2016). The Laplace

approximation has been designed to approximate integrals in the form

$$\int_{\mathcal{R}^{u_i}} \exp\{Q(\boldsymbol{u}_i)\}\mathrm{d}\boldsymbol{u}_i \approx (2\pi)^{n_u/2}\,|Q''(\hat{\boldsymbol{u}}_i)|^{-1/2}\,\exp\{Q(\hat{\boldsymbol{u}}_i)\}, \tag{2.4}$$

where $Q(\boldsymbol{u}_i)$ is a known, unimodal bounded function, and $\hat{\boldsymbol{u}}_i$ is the value for which $Q(\boldsymbol{u}_i)$ is maximized. As Wood (2015) shows, a Laplace approximation consists of a second order Taylor expansion of $\log f(\boldsymbol{y}_i, \boldsymbol{u}_i \mid \boldsymbol{\theta})$, about $\hat{\boldsymbol{u}}_i$, that gives

$$\log f(\boldsymbol{y}_i, \boldsymbol{u}_i \mid \boldsymbol{\theta}) \approx \log f(\boldsymbol{y}_i, \hat{\boldsymbol{u}}_i \mid \boldsymbol{\theta}) - \frac{1}{2}(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)^\top \boldsymbol{H}\,(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i),$$

where $\boldsymbol{H} = -\nabla^2_u \log f(\boldsymbol{y}_i, \hat{\boldsymbol{u}}_i \mid \boldsymbol{\theta})$. Hence, we can approximate the joint by

$$f(\boldsymbol{y}_i, \boldsymbol{u}_i \mid \boldsymbol{\theta}) \approx f(\boldsymbol{y}_i, \hat{\boldsymbol{u}}_i \mid \boldsymbol{\theta})\,\exp\left\{-\frac{1}{2}(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)^\top \boldsymbol{H}\,(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)\right\}. \tag{2.5}$$

From here we start to take advantage of the points mentioned above.

First, the fact that we are dealing with Gaussian distributed latent effects. In Equation 2.5 we have the core of a Gaussian density, that complete is

$$\int_{\mathcal{R}^{u_i}} \frac{1}{(2\pi)^{n_u/2}\,|\boldsymbol{H}^{-1}|^{1/2}}\,\exp\left\{-\frac{1}{2}(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)^\top \boldsymbol{H}\,(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)\right\}\mathrm{d}\boldsymbol{u}_i = 1$$

i.e., integrates to 1. Integrating Equation 2.5 follows that

$$\begin{aligned}\int_{\mathcal{R}^{u_i}} f(\boldsymbol{y}_i, \boldsymbol{u}_i \mid \boldsymbol{\theta})\,\mathrm{d}\boldsymbol{u}_i &\approx f(\boldsymbol{y}_i, \hat{\boldsymbol{u}}_i \mid \boldsymbol{\theta}) \int_{\mathcal{R}^{u_i}} \exp\left\{-\frac{1}{2}(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)^\top \boldsymbol{H}\,(\boldsymbol{u}_i - \hat{\boldsymbol{u}}_i)\right\}\mathrm{d}\boldsymbol{u}_i \\ &= (2\pi)^{n_u/2}\,|\boldsymbol{H}|^{-1/2}\,f(\boldsymbol{y}_i, \hat{\boldsymbol{u}}_i \mid \boldsymbol{\theta})\end{aligned}$$

i.e., we get Equation 2.4, a first order Laplace approximation to the integral. Careful accounting of the approximation error shows it to generally be $\mathcal{O}(n^{-1})$, where n is the sample size, and assuming a fixed length for $\boldsymbol{u}_i$ (WOOD, 2015).

The second advantage of a Laplace approximation approach in a GLMM is the exponential family structure. In a usual GLMM the response follows a one-parameter exponential family distribution that can be written as

$$f(\boldsymbol{y}_i \mid \boldsymbol{u}_i, \boldsymbol{\theta}) = \exp\left\{\boldsymbol{y}_i^\top(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{u}_i) - \mathbf{1}_i^\top b(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{u}_i) + \mathbf{1}_i^\top c(\boldsymbol{y}_i)\right\},$$

where $b(\cdot)$ and $c(\cdot)$ are known functions.

This general and easy to compute expression, together with a (multivariate) Gaussian distribution, highlights the convenience of the Laplace method. The $Q(\boldsymbol{u}_i)$ function to be maximized can be expressed as

$$\begin{aligned}Q(\boldsymbol{u}_i) = {} & \boldsymbol{y}_i^\top(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{u}_i) - \mathbf{1}_i^\top b(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{u}_i) + \mathbf{1}_i^\top c(\boldsymbol{y}_i) \\ & - \frac{n_u}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\boldsymbol{u}_i^\top \boldsymbol{\Sigma}^{-1}\,\boldsymbol{u}_i.\end{aligned} \tag{2.6}$$

The approximation in Equation 2.4 requires the maximum $\hat{\boldsymbol{u}}_i$ of the function $Q(\boldsymbol{u}_i)$. As we assume a Gaussian distribution with a known mean for the latent effects, we have the perfect initial guess for a gradient-based maximization method, as the Newton-Raphson (NR) algorithm.

The NR method consists of an iterative scheme as follows:

$$\boldsymbol{u}_i^{(k+1)} = \boldsymbol{u}_i^{(k)} - Q''(\boldsymbol{u}_i^{(k)})^{-1}\, Q'(\boldsymbol{u}_i^{(k)}), \quad k = 0,\, 1,\, \ldots$$

until convergence, which gives $\hat{\boldsymbol{u}}_i$. At this stage, all parameters $\boldsymbol{\theta}$ are considered known. Bonat & Ribeiro (2016) presents the generic expressions for the derivatives required by the NR method, given by the following:

$$\begin{aligned} Q'(\boldsymbol{u}_i^{(k)}) &= \{\boldsymbol{y}_i - b'(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{u}_i^{(k)})\}^\top - \boldsymbol{u}_i^{(k)\top}\boldsymbol{\Sigma}^{-1}, \\ Q''(\boldsymbol{u}_i^{(k)}) &= -\text{diag}\{b''(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\boldsymbol{u}_i^{(k)})\} - \boldsymbol{\Sigma}^{-1}. \end{aligned}$$

We have the initial guesses at $k = 0$.

Finally, the marginal log-likelihood function returned by the Laplace approximation, to each individual or unit under study i, is as follows:

$$\begin{aligned} l(\boldsymbol{\theta} \mid \boldsymbol{y}_i) = \log L(\boldsymbol{\theta} \mid \boldsymbol{y}_i) = \frac{n}{2}\log(2\pi) - \frac{1}{2}\log\left|\text{diag}\{b''(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\hat{\boldsymbol{u}}_i)\} + \boldsymbol{\Sigma}^{-1}\right| \\ + \boldsymbol{y}_i^\top(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\hat{\boldsymbol{u}}_i) - \mathbf{1}_i^\top b(\boldsymbol{x}_i\boldsymbol{\beta} + \boldsymbol{z}_i\hat{\boldsymbol{u}}_i) + \mathbf{1}_i^\top c(\boldsymbol{y}_i) \\ - \frac{n_{\boldsymbol{u}}}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\hat{\boldsymbol{u}}_i^\top\boldsymbol{\Sigma}^{-1}\,\hat{\boldsymbol{u}}_i, \end{aligned}$$

that can now be numerically maximized over the model parameters $\boldsymbol{\theta} = [\boldsymbol{\beta}\ \boldsymbol{\Sigma}]^\top$.

2.3 OPTIMIZATION: MARGINAL LIKELIHOOD FUNCTION

At this point it is already clear that we have two optimizations to be performed, an "inside" and an "outside" optimization. The inside one is made into the Laplace approximation layer via a Newton-Raphson algorithm, a Newton's method. The outside optimization is made with the Laplace approximation outputs i.e., the maximization of Equation 2.3's marginal log-likelihood over its parameters $\boldsymbol{\theta}$. This task is usually performed via a quasi-Newton method, we focus on two of the most traditional ones: the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm and the PORT routines.

The inside optimization is the numerical maximization of the joint log-likelihood with respect to (w.r.t.) its latent effects. This is kind of a simple task since all model parameters are considered as fixed, and we "know" that the latent effects are distributed with zero mean i.e., we have the perfect initial guess. In this context, the use of a Newton's method is straightforward. When we talk about the outside optimization

it is a completely different scenario, it is not straightforward to find a good initial guess or reach convergence. Thus, more robust methods are a good choice.

In optimization, Newton methods are algorithms for finding local maxima and minima of functions i.e., the search for the zeroes of the gradient of that function. Newton methods are characterized by the use of a symmetric matrix of function's second derivatives, the Hessian matrix. Quasi-Newton methods are based on Newton's method and are seen as an alternative to it. They can be used if the Hessian is unavailable or if is too expensive to compute it at every iteration.

As shown in Nocedal & Wright (2006), major advantages of quasi-Newton methods over Newton's method are that the Hessian matrix does not need to be computed, it is approximated; and it also does not need to be inverted. Newton's method requires the Hessian to be inverted, typically by solving a system of linear equations - often quite costly. In contrast, quasi-Newton methods usually generate an estimate of it directly. As in Newton's method, they use a second-order approximation to find the minimum of a function $f(\boldsymbol{x})$. The Taylor series of $f(\boldsymbol{x})$ around an iterate is

$$f(\boldsymbol{x}_k + \Delta\boldsymbol{x}) \approx f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^\top \Delta\boldsymbol{x} + \frac{1}{2}\Delta\boldsymbol{x}^\top \boldsymbol{B}\, \Delta\boldsymbol{x},$$

where $\nabla f(\cdot)$ is the gradient, and $\boldsymbol{B}$ an approximation to the Hessian matrix. The gradient of this approximation w.r.t. $\Delta\boldsymbol{x}$ is

$$\nabla f(\boldsymbol{x}_k + \Delta\boldsymbol{x}) \approx \nabla f(\boldsymbol{x}_k) + \boldsymbol{B}\, \Delta\boldsymbol{x},$$

setting this gradient to zero provides the Newton step:

$$\Delta\boldsymbol{x} = -\boldsymbol{B}^{-1}\nabla f(\boldsymbol{x}_k).$$

The Hessian approximation $\boldsymbol{B}$ is chosen to satisfy

$$\nabla f(\boldsymbol{x}_k + \Delta\boldsymbol{x}) = \nabla f(\boldsymbol{x}_k) + \boldsymbol{B}\, \Delta\boldsymbol{x},$$

which is called the *secant* equation i.e., the Taylor series of the gradient itself. Solving for $\boldsymbol{B}$ and applying the Newton's step with the updated value is equivalent to the *secant* method. Quasi-Newton methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems. The various quasi-Newton methods differ in their choice of the solution to the secant equation.

In a general quasi-Newton method, the unknown $\boldsymbol{x}_k$ is updated applying the Newton's step calculated using the current approximate Hessian matrix $\boldsymbol{B}_k$ in the following fashion:

- $\Delta\boldsymbol{x}_k = -\alpha_k \boldsymbol{B}_k^{-1}\nabla f(\boldsymbol{x}_k)$, with α chosen to satisfy the so called Wolfe conditions (NOCEDAL; WRIGHT, 2006, p. 34);

- $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta\boldsymbol{x}_k$;
- The gradient computed at the new point $\nabla f(\boldsymbol{x}_{k+1})$, and $\boldsymbol{y}_k = \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)$ is used to update the approximate Hessian $\boldsymbol{B}_{k+1}$, or directly its inverse $\boldsymbol{H}_{k+1} = \boldsymbol{B}_{k+1}^{-1}$.

The most popular quasi-Newton method is the BFGS algorithm, named for its discoverers Broyden, Fletcher, Goldfarb, and Shanno. It has the following update formula

$$\boldsymbol{B}_{k+1} = \boldsymbol{B}_k + \frac{\boldsymbol{y}_k \boldsymbol{y}_k^\top}{\boldsymbol{y}_k^\top \Delta\boldsymbol{x}_k} - \frac{\boldsymbol{B}_k \Delta\boldsymbol{x}_k (\boldsymbol{B}_k \Delta\boldsymbol{x}_k)^\top}{\Delta\boldsymbol{x}_k^\top \boldsymbol{B}_k \Delta\boldsymbol{x}_k},$$
$$\boldsymbol{H}_{k+1} = \boldsymbol{B}_{k+1}^{-1} = \left(\boldsymbol{I} - \frac{\Delta\boldsymbol{x}_k \boldsymbol{y}_k^\top}{\boldsymbol{y}_k^\top \Delta\boldsymbol{x}_k}\right) \boldsymbol{H}_k \left(\boldsymbol{I} - \frac{\boldsymbol{y}_k \Delta\boldsymbol{x}_k^\top}{\boldsymbol{y}_k^\top \Delta\boldsymbol{x}_k}\right) + \frac{\Delta\boldsymbol{x}_k \Delta\boldsymbol{x}_k^\top}{\boldsymbol{y}_k^\top \Delta\boldsymbol{x}_k}.$$

Another quasi-Newton method popular in the statistical modeling literature, is the one based on the PORT routines ⟨http://www.netlib.org/port/⟩. It is a Fortran mathematical subroutine library designed to be *portable* over different types of computers, developed by David Gay in the Bell Labs (GAY, 1990). Is a quasi-Newton adaptive nonlinear least-squares algorithm (DENNIS; GAY; WELSCH, 1981) with the following update formula

$$\begin{aligned}\boldsymbol{B}_{k+1} = \boldsymbol{B}_k &\\ &+ \frac{(\boldsymbol{y}_k - \boldsymbol{B}_k \Delta\boldsymbol{x}_k)\, \Delta\boldsymbol{x}_k^\top \boldsymbol{B}_k + \boldsymbol{B}_k \Delta\boldsymbol{x}_k\, (\boldsymbol{y}_k - \boldsymbol{B}_k \Delta\boldsymbol{x}_k)^\top}{\Delta\boldsymbol{x}_k^\top \boldsymbol{B}_k \Delta\boldsymbol{x}_k} \\ &- \frac{\Delta\boldsymbol{x}_k^\top\, (\boldsymbol{y}_k - \boldsymbol{B}_k \Delta\boldsymbol{x}_k)\, \boldsymbol{B}_k \Delta\boldsymbol{x}_k \Delta\boldsymbol{x}_k^\top \boldsymbol{B}_k}{\left(\Delta\boldsymbol{x}_k^\top \boldsymbol{B}_k \Delta\boldsymbol{x}_k\right)^\top \Delta\boldsymbol{x}_k^\top \boldsymbol{B}_k \Delta\boldsymbol{x}_k}.\end{aligned}$$

As Nocedal & Wright (2006) points out, each quasi-Newton method iteration can be performed at a cost of $\mathcal{O}(n^2)$ arithmetic operations (plus the cost of function and gradient evaluations); there are no $\mathcal{O}(n^3)$ operations such as linear system solves or matrix-matrix operations. In the BFGS algorithm is known that the rate of convergence is superlinear, which is a valid assumption to any quasi-Newton method and is fast enough for most practical purposes. Even though Newton's method converges more rapidly, quadratically, its cost per iteration usually is higher because of its need for second derivatives and solution of a linear system.

In this thesis, the used BFGS implementation is the one in the `R` (R Core Team, 2021) function `base::optim()`, and the PORT routine used is the one implemented in the `R` function `base::nlminb()`.

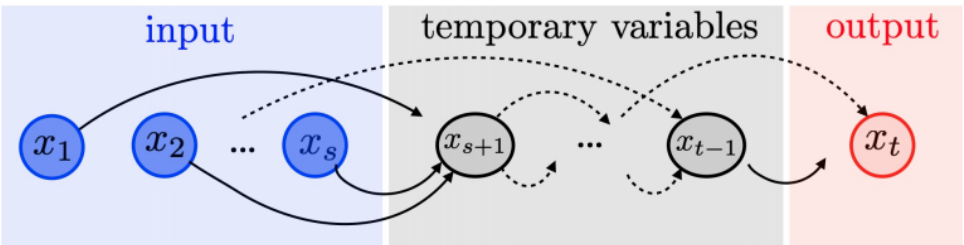
2.4 AD: AUTOMATIC DIFFERENTIATION

The computation of gradients, $\nabla f(\boldsymbol{x})$, are a fundamental and crucial task but also the main computational bottleneck to any Newton and quasi-Newton method. We choose to use the most efficient manner of computing gradients, and one of the best scientific computing techniques but still not so famous in the statistical modeling literature, the *automatic differentiation* (AD) procedure. AD has two modes, the so-called forward and reverse mode. We will talk a bit about both but we will use only the reverse mode. The reason can be illustraded by a simple example, given later.

Automatic differentiation, also called algorithmic differentiation or computational differentiation, is a set of techniques to numerically and recursively evaluate the derivative of a function specified by a computer program. AD techniques are based on the observation that any function, no matter how complicated, is evaluated by performing a sequence of simple elementary operations involving just one or two arguments at a time. Derivatives of arbitrary order can be computed automatically, automatized and accurately to working precision. Most of the information in this section was taken of Peyré (2020), but Wood (2015, p. 120) and Nocedal & Wright (2006, p. 204) are also very good references.

The most common differentiation approaches are finite differences (FD) and symbolic calculus. Considering a function $f : \mathbb{R}^p \to \mathbb{R}$ and the goal of deriving a method to evaluate $\nabla f : \mathbb{R}^p \to \mathbb{R}^p$, the approximation of this vector field via FD would require $p + 1$ evaluations of f. The same task via reverse mode AD has in most cases a cost proportional to a single evaluation of f. AD is similar to symbolic calculus in the sense that it provides an exact gradient computation, up to machine precision. However, symbolic calculus does not takes into account the underlying algorithm which compute the function, while AD factorizes the computation of the derivative according to an efficient algorithm. The use of AD is inherent to the use of a computational graph, as exemplified in Figure 2.

FIGURE 2 – A COMPUTATIONAL GRAPH



SOURCE: Peyré (2020, p. 31).

Assuming that f is implemented in an algorithm, the goal is to compute the

derivatives

$$\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}_k} \in \mathbb{R}^{n_t \times n_k},$$

for a numerical algorithm (succession of functions) of the form

$$\forall\, k = s+1,\, \ldots,\, t, \quad \boldsymbol{x}_k = f_k(\boldsymbol{x}_1,\, \ldots,\, \boldsymbol{x}_{k-1}),$$

where f_k is a function which only depends on the previous variables. The computational graph as in Figure 2, has the role of represent the linking of the variables involved in f_k to $\boldsymbol{x}_k$. The evaluation of $f(\boldsymbol{x})$ corresponds to a forward traversal of this graph.

Now, how we evaluate f through the graph? Via one of the AD modes.

2.4.1 Forward Mode

The forward mode correspond to the usual way of computing differentials. The method initialize with the derivative of the input nodes

$$\frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_1} = \mathrm{Id}_{n_1 \times n_1}, \quad \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1} = \mathbf{0}_{n_2 \times n_1}, \quad \frac{\partial \boldsymbol{x}_s}{\partial \boldsymbol{x}_1} = \mathbf{0}_{n_s \times n_1},$$

and then iteratively make use of the following recursion formula

$$\forall\, k = s+1,\, \ldots,\, t,$$
$$\frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{x}_1} = \sum_{l \,\in\, \text{father}(k)} \frac{\partial \boldsymbol{x}_k}{\partial \boldsymbol{x}_l} \times \frac{\partial \boldsymbol{x}_l}{\partial \boldsymbol{x}_1} = \sum_{l \,\in\, \text{father}(k)} \frac{\partial}{\partial \boldsymbol{x}_l} f_k(\boldsymbol{x}_1,\, \ldots,\, \boldsymbol{x}_{k-1}) \times \frac{\partial \boldsymbol{x}_l}{\partial \boldsymbol{x}_1}.$$

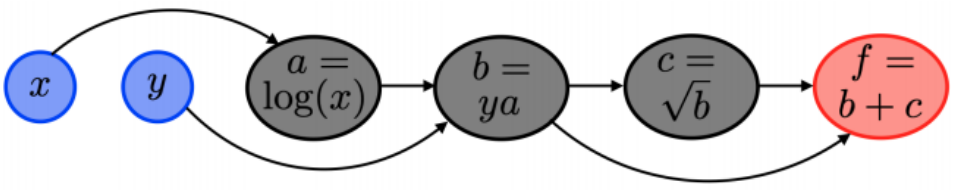
The notation "father(k)" denotes the nodes $l < k$ of the graph that are connected to k. We make use of Peyré (2020, p. 32)'s simple example.

Example. Consider the function

$$f(x,y) = y\log(x) + \sqrt{y\log(x)}$$

with the corresponding computational graph being displayed in Figure 3.

FIGURE 3 – EXAMPLE OF A SIMPLE COMPUTATIONAL GRAPH



SOURCE: Peyré (2020, p. 33).

The forward mode iterations to compute the derivative w.r.t. x following the computational graph, are given by

$$\begin{aligned}
&\frac{\partial x}{\partial x} = 1, \quad \frac{\partial y}{\partial x} = 0 \\
&\frac{\partial a}{\partial x} = \frac{\partial a}{\partial x}\frac{\partial x}{\partial x} = \frac{1}{x}\frac{\partial x}{\partial x} && \{x \mapsto a = \log(x)\} \\
&\frac{\partial b}{\partial x} = \frac{\partial b}{\partial a}\frac{\partial a}{\partial x} + \frac{\partial b}{\partial y}\frac{\partial y}{\partial x} = y\frac{\partial a}{\partial x} + 0 && \{(y,a) \mapsto b = ya\} \\
&\frac{\partial c}{\partial x} = \frac{\partial c}{\partial b}\frac{\partial b}{\partial x} = \frac{1}{2\sqrt{b}}\frac{\partial b}{\partial x} && \{b \mapsto c = \sqrt{b}\} \\
&\frac{\partial f}{\partial x} = \frac{\partial f}{\partial b}\frac{\partial b}{\partial x} + \frac{\partial f}{\partial c}\frac{\partial c}{\partial x} = 1\frac{\partial b}{\partial x} + 1\frac{\partial c}{\partial x} && \{(b,c) \mapsto f = b + c\}
\end{aligned}$$

To compute the derivative w.r.t. y we run another forward process

$$\begin{aligned}
&\frac{\partial x}{\partial y} = 0, \quad \frac{\partial y}{\partial y} = 1 \\
&\frac{\partial a}{\partial y} = \frac{\partial a}{\partial x}\frac{\partial x}{\partial y} = 0 && \{x \mapsto a = \log(x)\} \\
&\frac{\partial b}{\partial y} = \frac{\partial b}{\partial a}\frac{\partial a}{\partial y} + \frac{\partial b}{\partial y}\frac{\partial y}{\partial y} = 0 + a\frac{\partial y}{\partial y} && \{(y,a) \mapsto b = ya\} \\
&\frac{\partial c}{\partial y} = \frac{\partial c}{\partial b}\frac{\partial b}{\partial y} = \frac{1}{2\sqrt{b}}\frac{\partial b}{\partial y} && \{b \mapsto c = \sqrt{b}\} \\
&\frac{\partial f}{\partial y} = \frac{\partial f}{\partial b}\frac{\partial b}{\partial y} + \frac{\partial f}{\partial c}\frac{\partial c}{\partial y} = 1\frac{\partial b}{\partial y} + 1\frac{\partial c}{\partial y} && \{(b,c) \mapsto f = b + c\}
\end{aligned}$$

2.4.2 Reverse Mode

Instead of evaluating the differentials for all the input nodes, which is problematic for a large number of nodes, the reverse mode evaluates the differentials of the output node w.r.t. all the inner nodes.

The method is based on a backward adjoint chain rule and initialize with the derivative of the final node

$$\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{x}_t} = \mathrm{Id}_{n_t \times n_t},$$

and then from the last to the first node, iteratively make use of the following recursion formula

$$\begin{aligned}
&\forall\, k = t-1,\ t-2,\ \ldots,\ 1, \\
&\frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{x}_k} = \sum_{m \,\in\, \mathrm{son}(k)} \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{x}_m} \times \frac{\partial \boldsymbol{x}_m}{\partial \boldsymbol{x}_k} = \sum_{m \,\in\, \mathrm{son}(k)} \frac{\partial \boldsymbol{x}_t}{\partial \boldsymbol{x}_m} \times \frac{\partial}{\partial \boldsymbol{x}_k} f_m(\boldsymbol{x}_1,\ \ldots,\ \boldsymbol{x}_m).
\end{aligned}$$

The notation "son(k)" denotes the nodes $m < k$ of the graph that are connected to k. To be clear, the same simple example.

Example. Consider again the function

$$f(x,y) = y\log(x) + \sqrt{y\log(x)}.$$

The iterations of the reverse mode are given by

$$\begin{aligned}
&\frac{\partial f}{\partial f} = 1 \\
&\frac{\partial f}{\partial c} = \frac{\partial f}{\partial f}\frac{\partial f}{\partial c} = \frac{\partial f}{\partial f}1 && \{c \mapsto f = b + c\} \\
&\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c}\frac{\partial c}{\partial b} + \frac{\partial f}{\partial f}\frac{\partial f}{\partial b} = \frac{\partial f}{\partial c}\frac{1}{2\sqrt{b}} + \frac{\partial f}{\partial f}1 && \{b \mapsto c = \sqrt{b},\ b \mapsto f = b + c\} \\
&\frac{\partial f}{\partial a} = \frac{\partial f}{\partial b}\frac{\partial b}{\partial a} = \frac{\partial f}{\partial b}y && \{a \mapsto b = ya\} \\
&\frac{\partial f}{\partial y} = \frac{\partial f}{\partial b}\frac{\partial b}{\partial y} = \frac{\partial f}{\partial b}a && \{y \mapsto b = ya\} \\
&\frac{\partial f}{\partial x} = \frac{\partial f}{\partial a}\frac{\partial a}{\partial x} = \frac{\partial f}{\partial a}\frac{1}{x} && \{x \mapsto a = \log(x)\}
\end{aligned}$$

This is the advantage of reverse mode over the forward mode. A single traversal over the computational graph allows to compute both derivatives w.r.t. x and y, while the forward mode necessities two processes.

An drawback of the reverse mode is the need to store the entire computational graph, which is needed for the reverse sweep. In principle, storage of this graph is not too difficult to implement. However, the main benefit of AD is higher accuracy, and in many applications the cost is not critical.

2.5 TMB: TEMPLATE MODEL BUILDER

Note that the goal of AD is not to define an efficient computational graph, it is up to the user to provide it. However, computing an efficient graph associated to a mathematical formula is a complicated combinatorial problem. Thus, since our goal is to be able to fit our desired statistical models, a computational tool able to efficiently define and implement this computational graph is make necessary. To solve this and many other tasks we have the Template Model Builder (TMB), developed by Kristensen et al. (2016).

TMB ⟨http://tmb-project.org⟩ is an `R` (R Core Team, 2021) package for fitting statistical latent variable models to data, inpired by AD Model Builder (ADMB) (FOURNIER et al., 2012). ADMB is a statistical application for fitting nonlinear statistical models and solve optimization problems, that implements AD using `C++` classes and a native template language. Unlike most `R` packages, in TMB the model is formulated in `C++`. This characteristic provides great flexibility but requires some familiarity with the

C/C++ programming language. With TMB a user should be able to quickly implement complex latent effect models through simple C++ templates.

FIGURE 4 – EXAMPLE

```
1 frase <- 'palavras'
```

SOURCE: The author (2021).

In this chapter we describe step-by-step all the processes involved in the creation and parameter estimation of a GLMM. With TMB all this is put in practice in an efficient and robust fashion.

The user needs to provide just the negative joint log-likelihood function writing in a C++ template, using specialized macros that pass the parameters, latent effects and data from R. When the model presents latent effects, during the compilation the latent effects are integrated out via an efficient Laplace approximation routine with a Newton algorithm inside, and the negative marginal log-likelihood gradient is computed, via AD, and returned. The negative marginal log-likelihood is returned into an R object that can then be optimized using the user's favorite quasi-Newton routine, available in R. To accomplish all that, TMB combines some state-of-art software

- CppAD, a C++ AD package ⟨https://coin-or.github.io/CppAD/⟩;
- Eigen (GUENNEBAUD; JACOB et al., 2010), a C++ templated matrix-vector library;
- CHOLMOD, C sparse matrix routines available from R, used to obtain an efficient implementation of the Laplace approximation with exact derivatives ⟨https://developer.nvidia.com/cholmod⟩;
- Parallelism through BLAS ⟨http://www.netlib.org/blas/⟩, a Fortran tuned set of Basic Linear Algebra Subprograms;
- Matrix (BATES; MAECHLER, 2019), a rich hierarchy sparse and dense matrix classes and methods using LAPACK ⟨http://www.netlib.org/lapack/⟩ and SuiteSparse ⟨https://sparse.tamu.edu/⟩ libraries.

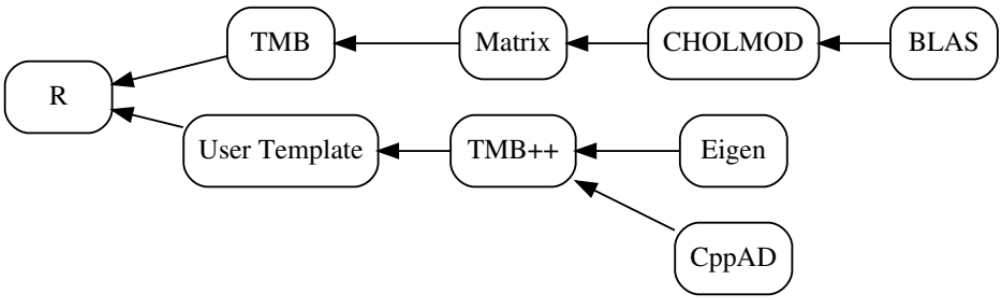
Also, some of its key characteristics are

- TMB employs AD to calculate first and second order derivatives of the log-likelihood function or of any objective function written in C++;
- The objective function, and its derivatives, can be called from R. Hence, parameter estimation via base::optim() or base::nlminb() is easy to be performed;

- Standard deviations of any parameter, or derived parameter, can be obtained via the *delta method* (Ver HOEF, 2012).

An overview of the pachage design is shown in Figure 5.

FIGURE 5 – TMB PACKAGE DESIGN



SOURCE: Kristensen et al. (2016).

Here we focus on GLMMs, but basically any statistical model with a latent structure (or not), linear (or not), can be fitted with TMB. In times of *big data* and with the TMB's authors having a professional preference for state-space and spatial models, TMB has also automatic sparseness detection.and some other nice built tools. Pre and post-processing of data should be done in R.

A TMB Users' mailing list exists, and it is extremely helpful for taking doubts and questions ⟨https://groups.google.com/g/tmb-users⟩. Also, a very didactic and comprehensive documentation with several examples is available online ⟨https://kaskr.github.io/adcomp/_book/Tutorial.html⟩.

3 multiGLMM: A MULTINOMIAL GLMM FOR CLUSTERED COMPETING RISKS DATA

The clustered competing risks setting is a complex and specific survival data structure. Although, we are using a general statistical modeling framework, a generalized linear mixed model (GLMM). Consequently, some specific features are necessary into the model construction to properly accommodate all the characteristics of the data structure.

To model competing risks data we need a multivariate model (several responses, causes of failure) and to choose in which scale to work on. We may work on the hazard scale and deal with the cause-specific hazard function or on the probability scale and deal with the cause-specific cumulative incidence function (CIF). Using the correct link function, we are able to construct an appropriate multivariate GLMM to work on the probability scale.

Our goal is to be able to deal with complex family studies, where there is generally a strong interest in describing age at disease onset in the scenarios of within-cluster dependence. The distribution of age at disease onset is directly described by the cause-specific CIF. To build a multivariate GLMM for this type of data we need to accommodate the cause-specific CIFs and the censorings. Assuming the conditional distribution for our model response as multinomial (a multivariate distribution) we already deal with both left-truncation and right-censoring, avoiding the specification of a censoring distribution. The cause-specific CIFs can be modeled via the link function of our, then, multinomial GLMM (multiGLMM). The multinomial distribution also guarantees that the CIFs of all causes are modeled.

Our choice for a general framework tries to make the inference of this complex model, easier. Besides, taking advantage of all the computational procedures mentioned in the previous chapter. This chapter presents our multiGLMM for clustered competing risks data and is divided into two sections. In Section 3.1 we discuss in detail the cluster-specific cumulative incidence function (CIF) and in Section 3.2 we present the complete modeling framework.

3.1 CLUSTER-SPECIFIC CUMULATIVE INCIDENCE FUNCTION (CIF)

Consider that the observed follow-up time of an individual is given by $T = \min(T^*, C)$, where T^* denote the failure time and C denote the censoring time. Given the possible covariates $\boldsymbol{x}$, for a cause-specific of failure k the cumulative incidence

function (CIF) is defined as

$$\begin{aligned} F_k(t \mid \boldsymbol{x}) &= \mathbb{P}[T \leq t,\, K = k \mid \boldsymbol{x}] \\ &= \int_0^t f_k(z \mid \boldsymbol{x})\, \mathrm{d}z \\ &= \int_0^t \lambda_k(z \mid \boldsymbol{x})\, S(z \mid \boldsymbol{x})\, \mathrm{d}z, \quad t > 0, \quad k = 1, \ldots, K. \end{aligned}$$

where $f_k(t \mid \boldsymbol{x})$ is the (sub)density for the time to a type k failure. This is the general definition of a CIF, and to define it we need to define the functions that compose the subdensity. The first is the cause-specific hazard function or process

$$\lambda_k(t \mid \boldsymbol{x}) = \lim_{h \to 0} \frac{1}{h}\, \mathbb{P}[t \leq T < t + h,\, K = k \mid T \geq t,\, \boldsymbol{x}], \quad t > 0, \quad k = 1, \ldots, K.$$

In words, the cause-specific hazard function, $\lambda_k(t \mid \boldsymbol{x})$, represents the instantaneous rate for failures of type k at time t given $\boldsymbol{x}$ and all other failure types (competing causes). If we sum up all cause-specific hazard function we get the overall hazard function,

$$\lambda(t \mid \boldsymbol{x}) = \sum_{k=1}^{K} \lambda_k(t \mid \boldsymbol{x}).$$

From the overall hazard function we arrive in the overall survival function,

$$S(t \mid \boldsymbol{x}) = \mathbb{P}[T > t \mid \boldsymbol{x}] = \exp\left\{ - \int_0^t \lambda(z \mid \boldsymbol{x})\, \mathrm{d}z \right\},$$

the second function that compose the subdensity $f_k(t \mid \boldsymbol{x})$. A comprehensive reference for all these definitions is the book of Kalbfleisch & Prentice (2002).

Until this point, we were talking about a general CIF's definition. We need now a precise framework telling how to take into consideration our clustered/family structure. We use the same CIF specification of Cederkvist et al. (2019), i.e. the approach that motivated this thesis. For two competing causes of failure, the cause-specific CIFs are specified in the following manner,

$$F_k(t \mid \boldsymbol{x},\, u_1,\, u_2,\, \eta_k) = \underbrace{\pi_k(\boldsymbol{x},\, u_1,\, u_2)}_{\substack{\text{cluster-specific} \\ \text{risk level}}} \times \underbrace{\Phi[w_k g(t) - \boldsymbol{x}\boldsymbol{\gamma}_k - \eta_k]}_{\substack{\text{cluster-specific} \\ \text{failure time trajectory}}}, \quad t > 0, \quad k = 1, 2. \tag{3.1}$$

i.e. as a product of a cluster-specific risk level and a cluster-specific failure time trajectory, resulting in a cluster-specific CIF. What makes these components cluster-specific are $\boldsymbol{u} = \{u_1,\, u_2\}$ and $\boldsymbol{\eta} = \{\eta_1,\, \eta_2\}$, Gaussian distributed latent effects with zero mean and potentially correlated, i.e.

$$\begin{bmatrix} u_1 \\ u_2 \\ \eta_1 \\ \eta_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{u_1} & \operatorname{cov}(u_1,\, u_2) & \operatorname{cov}(u_1,\, \eta_1) & \operatorname{cov}(u_1,\, \eta_2) \\ & \sigma^2_{u_2} & \operatorname{cov}(u_2,\, \eta_1) & \operatorname{cov}(u_2,\, \eta_2) \\ & & \sigma^2_{\eta_1} & \operatorname{cov}(\eta_1,\, \eta_2) \\ & & & \sigma^2_{\eta_2} \end{bmatrix} \right).$$

The cluster-specific survival function is given as $S(t \mid \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\eta}) = 1 - F_1(t \mid \boldsymbol{x}, \boldsymbol{u}, \eta_1) - F_2(t \mid \boldsymbol{x}, \boldsymbol{u}, \eta_2)$. Since we use the same CIF specification of Cederkvist et al. (2019), the following details are essentially the same encountered in the paper.

Focusing first on the second component of Equation 3.1, the cluster-specific failure time trajectory

$$\Phi[w_k g(t) - \boldsymbol{x}\gamma_k - \eta_k], \quad t > 0, \quad k = 1, 2,$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard Gaussian distribution. Instead of $w_k g(t)$, in Cederkvist et al. (2019) is specified $\alpha_k(g(t))$ with $\alpha_k(\cdot)$ being a monotonically increasing function known up to a finite-dimensional parameter vector, w_k. Examples are monotonically increasing B-splines or piecewise linear functions. However, to simplify the model structure we consider just the finite-dimensional parameter vector. The bottom line is that the authors do the same approach in their applications. With regard to the function $g(t)$, it plays a crucial role since the CIF separation in Equation 3.1 is only possible with it. A time t transformation given by

$$g(t) = \operatorname{arctanh}\left(\frac{t - \delta/2}{\delta/2}\right), \quad t \in (0, \delta), \quad g(t) \in (-\infty, \infty),$$

where δ depends on the data and cannot exceed the maximum observed follow-up time τ, i.e. $\delta \leq \tau$. With this Fisher-based transformation the value of the cluster-specific failure time trajectory is equal 1, at time δ. Consequently, $F_k(\delta \mid \boldsymbol{x}, \boldsymbol{u}, \eta_k) = \pi_k(\boldsymbol{x} \mid \boldsymbol{u})$ and we can interpret $\pi_1(\boldsymbol{x} \mid \boldsymbol{u})$ and $\pi_2(\boldsymbol{x} \mid \boldsymbol{u})$ as the cause-specific cluster-specific risk levels, at time δ.

The cluster-specific risk levels are modeled by a multinomial logistic regression model with latent effects, i.e.

$$\pi_k(\boldsymbol{x}, \boldsymbol{u}) = \frac{\exp\{\boldsymbol{x}\boldsymbol{\beta}_k + u_k\}}{1 + \exp\{\boldsymbol{x}\boldsymbol{\beta}_1 + u_1\} + \exp\{\boldsymbol{x}\boldsymbol{\beta}_2 + u_2\}}, \quad k = 1, 2. \tag{3.2}$$

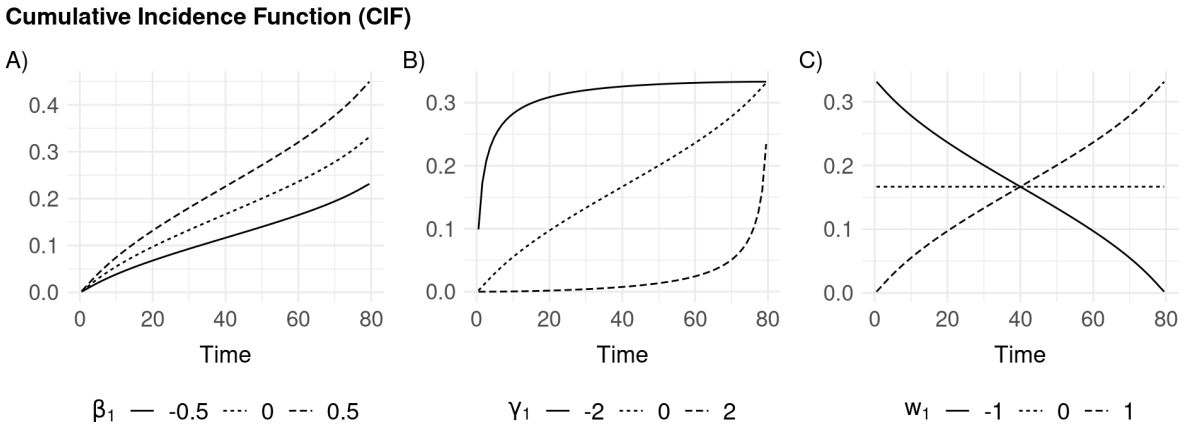
where the $\boldsymbol{\beta}_k$'s are the coefficients responsible for quantifying the impact of the covariates in the cause-specific risk levels. For individuals from the same chuster/family, at the same time point, the $\boldsymbol{\beta}_k$s have the well-known odds ratio interpretation.

A direct understanding of all coefficients/parameters of Equation 3.1 can be reached via the illustrations in Figure 6. To really understand what is going on, we simplify the model. We still consider just two competing causes but without covariates and we plot just the cluster-specific CIF of one failure cause. In Figure 6 A) we see that the β's are also related with the curve's maximum value, i.e. bigger the β, highest the CIF will reach.

The γ_k's are the coefficients responsible for quantifying the impact of the covariates in the cause-specific failure time trajectories, i.e. the shape of the cumulative

incidence. In Figure 6 B) we see that the γ's are also related with an idea of midpoint and consequently, growth speed. The fact that γ_k enters negatively in the cluster-specific failure time trajectory makes that a negative value causes an advance towards the curve, whereas a positive value causes a delay. By last but not least, the w's in Figure 6 C). With negative values, we have a decreasing curve and with positive values an increasing curve, i.e. we are interested only on the positive side.

FIGURE 6 – ILLUSTRATION OF COEFFICIENT BEHAVIORS FOR A GIVEN CUMULATIVE INCIDENCE FUNCTION (CIF), PROPOSED BY Cederkvist et al. (2019), IN A MODEL WITH TWO COMPETING CAUSES OF FAILURE, WITHOUT COVARIATES AND THE FOLLOWING CONFIGURATION: $\beta_2 = 0$, $\boldsymbol{u} = 0$ AND $\boldsymbol{\eta} = 0$; IN EACH SCENARIO ALL OTHER COEFFICIENTS ARE SET AT ZERO, WITH THE EXCEPTION OF $w_1 = 1$



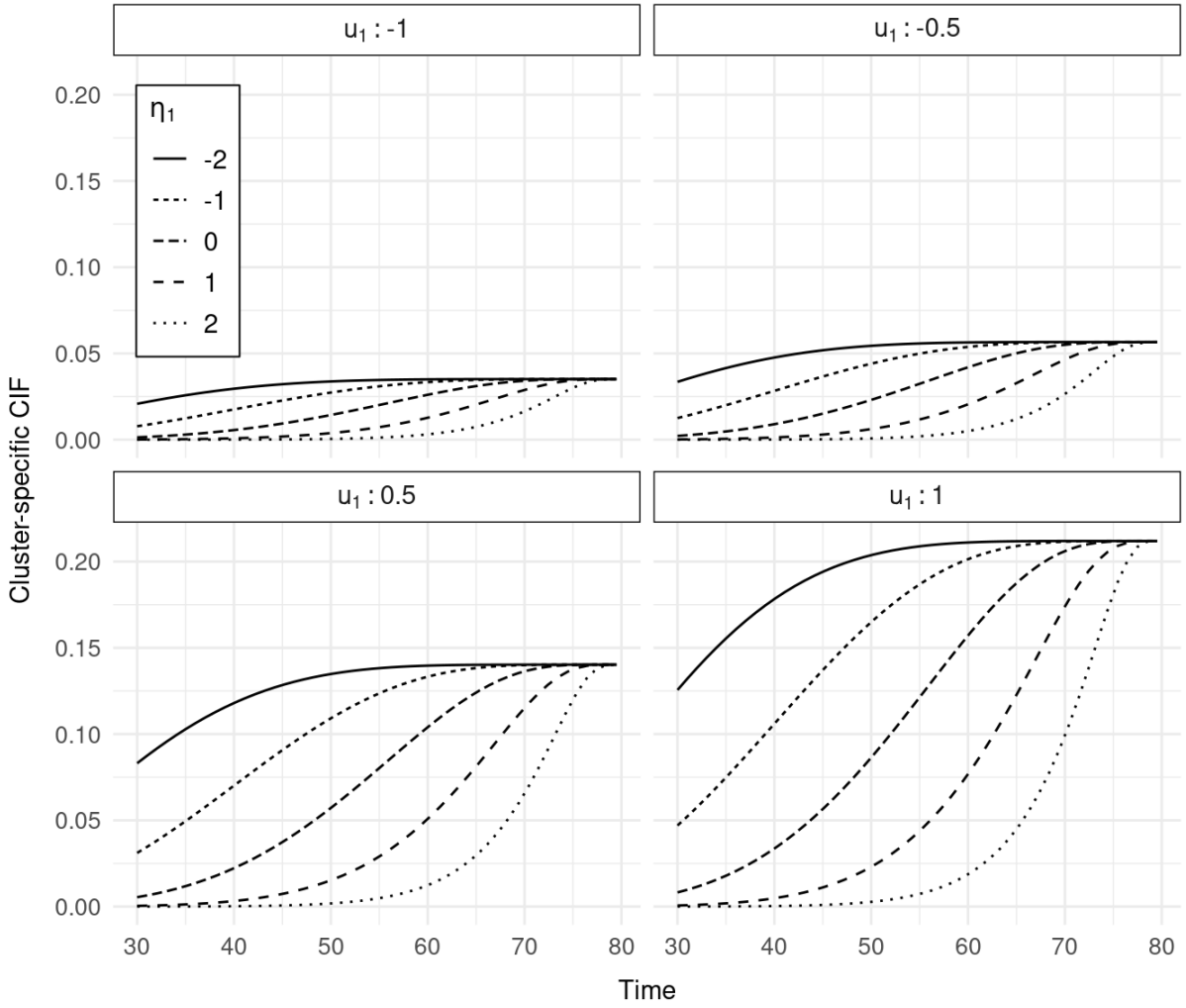
SOURCE: The author (2021).

Remains to talk about the within-cluster dependence induced by the latent effects in $\boldsymbol{u}$ and $\boldsymbol{\eta}$. Unfortunately, they do not have an easy interpretation. To help in the discussion, Figure 7 illustrates the cluster-specific CIF for a given failure cause in a model without covariates, let's call it failure cause 1 (in total we have two).

The latent effects u_1 and u_2 always appear together in the cluster-specific risk level, as consequency they have a joint effect on the cumulative incidence of both causes. Nevertheless, as we can see in Figure 7, an increase in u_k will increase the risk of failure from cause k and vice versa. The interpretation of $\text{cov}(\eta_1, \eta_2)$ and $\text{cov}(u_1, u_2)$ is straightforward, with regard to $\text{cov}(u_k, \eta_k)$ we can not say the same. A negative correlation between η_k and u_k imply that when η_k decreases, u_k increases and conversely when η_K increases, u_k decreases. In other words, an increased risk level is reached quickly and a decreased risk level is reached later, respectively.

Practical situations with a positive within-cause correlation are hard to find, i.e. where an increased risk level is associated with a late onset and vice versa. However, a positive cross-cause correlation between η and u sounds more realistic. i.e. where late onset of one failure cause is associated with a high absolute risk of another failure cause.

FIGURE 7 – ILLUSTRATION OF A GIVEN CLUSTER-SPECIFIC CUMULATIVE INCIDENCE FUNCTION (CIF), PROPOSED BY Cederkvist et al. (2019), IN A MODEL WITH TWO COMPETING CAUSES OF FAILURE, WITHOUT COVARIATES AND FOLLOWING CONFIGURATION: $\beta_1 = -2$, $\beta_2 = -1$, $\gamma_1 = 1$, $w_1 = 3$ AND $u_2 = 0$. THE VARIATION BETWEEN FRAMES IS GIVEN BY THE LATENT EFFECTS u_1 AND η_1



SOURCE: The author (2021).

The latent effects $\{u_k, \eta_k\}$ are assumed independent across clusters and shared by individuals within the same cluster/family.

3.2 MODEL SPECIFICATION

The multiGLMM for clustered competing risks data is specified in the following hierarchical fashion. By simplicity, we focus on two competing causes of failure but an extension is straightforward.

For two competing causes of failure, a subject i, in the cluster/family j, in time

t, we have

$$y_{ijt} \mid \{u_{1j},\, u_{2j},\, \eta_{1j},\, \eta_{2j}\} \sim \text{Multinomial}(p_{1ijt},\, p_{2ijt},\, p_{3ijt})$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \eta_1 \\ \eta_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{u_1} & \text{cov}(u_1,\, u_2) & \text{cov}(u_1,\, \eta_1) & \text{cov}(u_1,\, \eta_2) \\ & \sigma^2_{u_2} & \text{cov}(u_2,\, \eta_1) & \text{cov}(u_2,\, \eta_2) \\ & & \sigma^2_{\eta_1} & \text{cov}(\eta_1,\, \eta_2) \\ & & & \sigma^2_{\eta_2} \end{bmatrix} \right)$$

$$\begin{aligned} p_{kijt} &= \frac{\partial}{\partial t} F_k(t \mid \boldsymbol{x},\, u_1,\, u_2,\, \eta_k) \\ &= \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{1 + \sum_{m=1}^{K-1} \exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}} \\ &\quad \times w_k \frac{\delta}{2\delta t - 2t^2}\, \phi\left(w_k \text{arctanh}\left(\frac{t - \delta/2}{\delta/2} \right) - \boldsymbol{x}_{kij}\gamma_{ki} - \eta_{kj} \right), \end{aligned} \tag{3.3}$$

$k = 1,\, 2.$

The probabilities are given by the derivative w.r.t. time t of the cluster-specific CIF. The choice of a multinomial logistic regression model ensures that the sum of the predicted cause-specific CIFs does not exceed 1.

Considering two competing causes of failure, we have a multinomial with three classes. The third class exists to handle the censorship and its probability is given by the complementary to reach 1. This framework in Equation 3.3 results in what we call multiGLMM, a multinomial GLMM, to handle the CIF of clustered competing risks data. For a random sample, the corresponding marginal likelihood functions in given by

$$\begin{aligned} L(\boldsymbol{\theta}\,;\, y) &= \prod_{j=1}^{J} \int_{\Re^4} \pi(y_j \mid \boldsymbol{r}_j) \times \pi(\boldsymbol{r}_j)\, \mathrm{d}\boldsymbol{r}_j \\ &= \prod_{j=1}^{J} \int_{\Re^4} \underbrace{\left\{ \prod_{i=1}^{n_j} \prod_{t=1}^{n_{ij}} \left(\frac{(\sum_{k=1}^{K} y_{kijt})!}{y_{1ijt}!\, y_{2ijt}!\, y_{3ijt}!} \prod_{k=1}^{K} p_{kijt}^{y_{kijt}} \right) \right\}}_{\text{fixed effect component}} \times \\ &\qquad \underbrace{(2\pi)^{-2} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} \boldsymbol{r}_j^\top \Sigma^{-1} \boldsymbol{r}_j \right\}}_{\text{latent effect component}} \mathrm{d}\boldsymbol{r}_j \\ &= \prod_{j=1}^{J} \int_{\Re^4} \underbrace{\left\{ \prod_{i=1}^{n_j} \prod_{t=1}^{n_{ij}} \prod_{k=1}^{K} p_{kijt}^{y_{kijt}} \right\}}_{\text{fixed effect}} \underbrace{(2\pi)^{-2} |\Sigma|^{-1/2} \exp\left\{ -\frac{1}{2} \boldsymbol{r}_j^\top \Sigma^{-1} \boldsymbol{r}_j \right\}}_{\text{latent effect component}} \mathrm{d}\boldsymbol{r}_j, \end{aligned} \tag{3.4}$$

where $\boldsymbol{\theta} = [\boldsymbol{\beta}\ \boldsymbol{\gamma}\ \boldsymbol{w}\ \boldsymbol{\sigma}^2\ \boldsymbol{\rho}]^\top$ is the parameters vector to be maximized. In our framework, a subject can fail from just one competing cause or get censor, at a given time. Thus,

the fraction of factorials in the fixed effect component is made only by 0's and 1's. Finally, returning the value 1. The matrix Σ is the variance-covariance matrix, which components are given by σ^2 and ρ.

Now, Equation 3.4 in words. To each cluster (family) j we have a product of two components. The fixed effect component, given by a multinomial distribution with its probabilities specified through the cluster-specific CIF (Equation 3.1) and, the latent effect component, given by a multivariate Gaussian distribution.

To each subject i that composes a cluster j we have its specific fixed effects contribution. The likelihood in Equation 3.4 is the most general as possible, allowing for repeated measures to each subject. Since all subjects of a given cluster shares the same latent effect, we have just one latent effect contribution multiplying the product of fixed effect contributions. As we do not observe the latent effect variables, $\boldsymbol{r}_j$, we integrate out in it. With two competing causes of failure, we have four latent effects (a multivariate Gaussian distribution in four dimensions). Consequently, for each cluster, we approximate an integral in four dimensions. The product of these approximated integrals results in the called marginal likelihood, to be maximized in $\boldsymbol{\theta}$.

3.2.1 Parametrization

We have to choose in which terms we parameterize the variance-covariance matrix Σ. Besides the latent effects variances $\{\sigma^2\}$, we have to choose if we will estimate its covariances or correlations. By the name *variance-covariance* matrix, it is natural to think on covariance terms. However, this option is not very attractive since its interpretation is not clear. A more attractive choice is in terms of correlation.

The covariance between two terms is defined as a triple product: the two terms standard deviations times the correlation, ρ. Still thinking in two competing causes of failure, we have an Σ matrix with six correlations

$$\Sigma = \begin{bmatrix} \sigma^2_{u_1} & \rho_{u_1,u_2}\,\sigma_{u_1}\sigma_{u_2} & \rho_{u_1,\eta_1}\,\sigma_{u_1}\sigma_{\eta_1} & \rho_{u_1,\eta_2}\,\sigma_{u_1}\sigma_{\eta_2} \\ & \sigma^2_{u_2} & \rho_{u_2,\eta_1}\,\sigma_{u_2}\sigma_{\eta_1} & \rho_{u_2,\eta_2}\,\sigma_{u_2}\sigma_{\eta_2} \\ & & \sigma^2_{\eta_1} & \rho_{\eta_1,\eta_2}\,\sigma_{\eta_1}\sigma_{\eta_2} \\ & & & \sigma^2_{\eta_2} \end{bmatrix}.$$

With the matrix parametrization being chosen, we have that the parameters to be estimated are the components of the vector $\boldsymbol{\theta} = [\boldsymbol{\beta}\ \boldsymbol{\gamma}\ \boldsymbol{w}\ \sigma^2\ \boldsymbol{\rho}]^\top$. There we have the fixed effects or mean components $\{\boldsymbol{\beta}\ \boldsymbol{\gamma}\ \boldsymbol{w}\}$, the easiest to estimate in a statistical modeling framework; we have variance components $\{\sigma^2\}$, the intermediate ones; and the correlation components $\{\boldsymbol{\rho}\}$, the hardest ones. This idea of easy or hard to estimate may be justified by three, connected, arguments.

The first comes from the fact that we are modeling the mean of a probability distribution in a hierarchical and structured fashion, consequently, the easiest parameters to estimate will be the mean components. We may make the analogy that to estimate the mean parameters we need data (resources); to estimate the variance parameters we need more data (more resources), and to estimate the correlation parameters we need much more data (even more resources). The second argument comes to also explain the first one via the parametric space constraints.

Generally, the fixed effect components do not present constraints, i.e. they can vary in all $\mathbb{R}$. The same can not be said from the variance components, constrained by definition into the $\mathbb{R}_*^+$. Finally, we have the correlation components, constrained to the interval $[-1, 1]$. These parametric space constraints drive us again to the first argument since we need more data (resources, information) to be able to estimate coefficients constrained to some interval. Nevertheless, this may not be enough. Without providing some extra information, in terms of an constrained algorithm e.g., it is very reasonable to expect that during the optimization procedure some unrealistic areas of the parametric space could be visited and jeopardize the stability or even the whole optimization procedure. To overcome these possible difficulties, parameter reparametrizations are more than welcome.

The variance and correlation parameters are modeled in terms of the matrix Σ. This matrix is symmetric and more important, positive semi-definite. This last characteristic is also the third argument to justify why is so difficult to estimate these parameters. Since the estimates should lead to a positive semi-definite matrix, the employment of a parametrization is welcome to enforces this condition.

In the subject of choosing the components parametrization for a positive-definite matrix Σ, we have basically two big options available in the statistical modeling literature. One of them consists of just transform the scale. By practical reasons, let us think in a 2×2 matrix

$$\Sigma = \begin{bmatrix} \exp\{\log \sigma_1^2\} & z^{-1}(z(\rho_{1,2}))\sqrt{\exp\{\log \sigma_1^2\}}\sqrt{\exp\{\log \sigma_1^2\}} \\ & \exp\{\log \sigma_2^2\} \end{bmatrix},$$

i.e. in the main diagonal we now estimate the log variances and in the off-diagonal we estimate Fisher z-transformed correlations.

The estimation of the log variances has two big advantages

- Since the natural lorarithm is a real-valued function, we overcome the parametric space constraint problem;
- High variances are problematic for many reasons but in the context of seeing them as the diagonal components of a restricted matrix, being able to control its

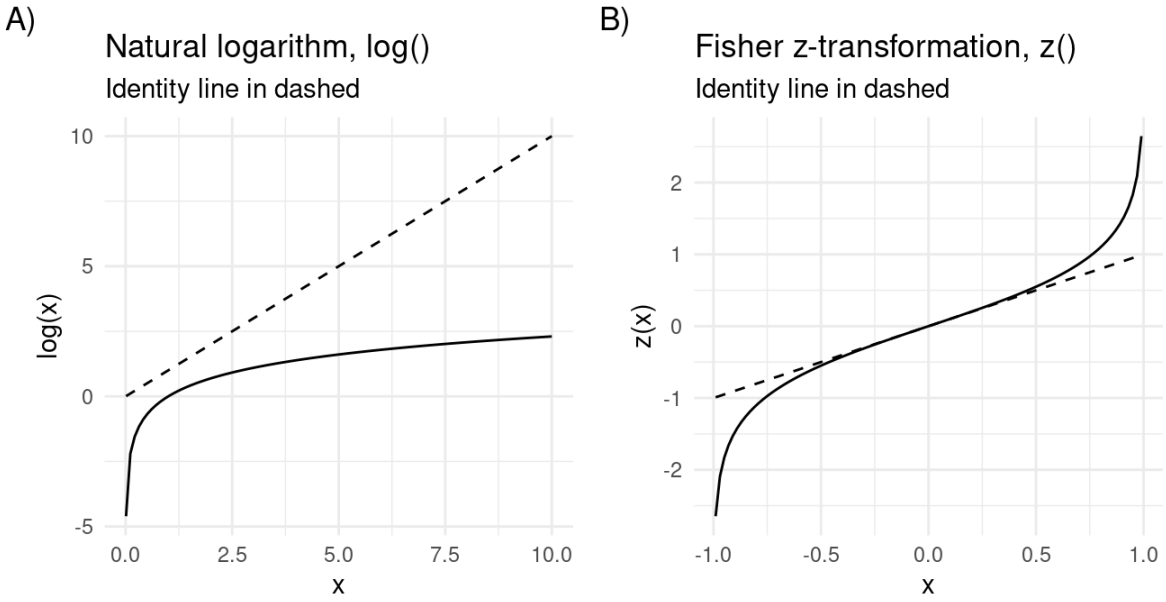
magnitudes is a crucial task to the stability of any optimization routine. With the natural logarithm transformation we shrink the parametric space, as illustrated in Figure 8 A), avoiding some eventual numerical cumbersome.

With the correlation components what we do is to proceed with the estimation of its Fischer z-transformation. This transformation, and its inverse, are defined as

$$z(\rho) = \frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right) = \operatorname{arctanh}(\rho), \quad z^{-1}(\rho) = \frac{\exp\{2\rho\}-1}{\exp\{2\rho\}+1} = \tanh(\rho).$$

The Fisher z-transformation plays the role here of stretching the small correlation parametric space but doing this in a smooth fashion, as illustrated in Figure 8 B).

FIGURE 8 – ILLUSTRATION OF THE PARAMETRIZATION BEHAVIOR FOR THE VARIANCE COMPONENTS, IN A), AND CORRELATION COMPONENTS, IN B)



SOURCE: The author (2021).

The other parametrization option consist in estimate the elements of a factorization or decomposition of the positive-definite matrix Σ. The most common is the Cholesky factorization or decomposition (PINHEIRO; BATES, 1996). For two competing causes of failure, a standard Cholesky decomposition of Σ may be expressed as

$$\Sigma = \begin{bmatrix} c_1 & 0 & 0 & 0 \\ c_2 & c_3 & 0 & 0 \\ c_4 & c_5 & c_6 & 0 \\ c_7 & c_8 & c_9 & c_{10} \end{bmatrix} \begin{bmatrix} c_1 & c_2 & c_4 & c_7 \\ 0 & c_3 & c_5 & c_8 \\ 0 & 0 & c_6 & c_9 \\ 0 & 0 & 0 & c_{10} \end{bmatrix} = LL^\top,$$

where $\{c_i\}_{i=1}^{10}$ are then the coefficients to be estimated.

A disadvantage in the use of a decomposition as the Cholesky is the lack of a straightforward interpretation to the elements $\{c_i\}_{i=1}^{10}$. However, with the application of the delta method, already implemented in TMB (KRISTENSEN et al., 2016), it is straightforward to get back the Σ elements together with its respective standard errors. The main advantage of this parametrization, apart from the fact that it ensures positive definiteness, is that it is computationally simple and stable.

Just to mention another viable possibilities, we could use a modified Cholesky decomposition (POURAHMADI, 2007) that provides a better statistical interpretation of the decomposition elements or, we could also parametrize the precision matrix, $\boldsymbol{Q} = \Sigma^{-1}$. Since we use Σ^{-1} in the marginal likelihood of Equation 3.4, parametrizing directly its inverse save us some computations.

Besides the popularity of the Cholesky method, there is another factorization scheme available and efficiently implemented in TMB. It is a factorization based on a vector scale transformation of an unstructured correlation matrix. For two competing causes of failure the decomposition is specified in the following fashion

$$\Sigma = VD^{-1/2}LL^{\top}D^{-1/2}V^{\top},$$

where

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ c_1 & 1 & 0 & 0 \\ c_2 & c_3 & 1 & 0 \\ c_4 & c_5 & c_6 & 1 \end{bmatrix}, \quad D = \text{diag}(LL^{\top}) \quad \text{and} \quad W = \text{diag}(\{\sigma_i\}_{i=1}^{4}).$$

This scheme is based initially on the factorization of a correlation matrix (unit diagonal) as $D^{-1/2}LL^{\top}D^{-1/2}$. The elements $\{c_i\}_{i=1}^{6}$ to be estimated has the advantage of being unconstrained and guarantees that the symmetry and positive definiteness constraint is respected. The variances are scaled via the diagonal matrix V, its elements $\{\sigma_i\}_{i=1}^{4}$ are then the standard deviations to be estimated.

4 DATASETS

This chapter describes how to simulate from our multiGLMM, and describes a real-based dataset used as an application example. The simulation procedure is addressed in Section 4.1. In Section 4.2 a simulated dataset based on the Nordic Cancer Union (NCU) twins data is presented as an application example.

4.1 SIMULATING FROM THE MODEL

Being able to simulate data from a model is a key task, fundamental to assess the finite-sample properties and the estimation procedure liability of a given statistical model. The step-by-step describing the simulation procedure of our multiGLMM is presented on Algorithm 1, following the model hierarchical structure stipulated in Equation 3.3.

ALGORITHM 1 SIMULATING FROM A multiGLMM FOR CLUSTERED COMPETING RISKS DATA

1: Set J, the number of clusters
2: Set n_j, the number of cluster elements ▷ can be of different sizes
3: Set $K-1$, the number of competing causes of failure
4: Set the model parameter values $\boldsymbol{\theta} = [\boldsymbol{\beta}\ \boldsymbol{\gamma}\ \boldsymbol{w}\ \sigma^2\ \boldsymbol{\varrho}]^\top$
5: Sample J latent effect vectors from a $\mathcal{N}_{(K-1)\times(K-1)}(\mathbf{0},\ \Sigma(\sigma^2, \boldsymbol{\varrho}))$
6: Set δ ▷ maximum follow-up time
7: Set the failure times t_{ij}
8: Compute the competing risks probabilities

$$p_{kij} = \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{1 + \sum_{m=1}^{K-1}\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}} \times w_k \frac{\delta}{2\delta t_{ij} - 2t_{ij}^2}\, \phi\left(w_k \operatorname{arctanh}\left(\frac{t_{ij} - \delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}\right),$$

$$\text{Censorship}: \quad p_{Kij} = 1 - \sum_{k=1}^{K-1} p_{kij}, \quad k = 1, 2, \ldots, K-1$$

9: Sample $J \times n_j$ vectors from a Multinomial$(p_{1ij},\ p_{2ij},\ \ldots,\ p_{Kij})$
10: If $t_{ij} = \delta$, moves to class K ▷ any failure at time δ is a censorship
11: **return** Multinomial vectors and their respective failure/censoring times

SOURCE: The author (2021).

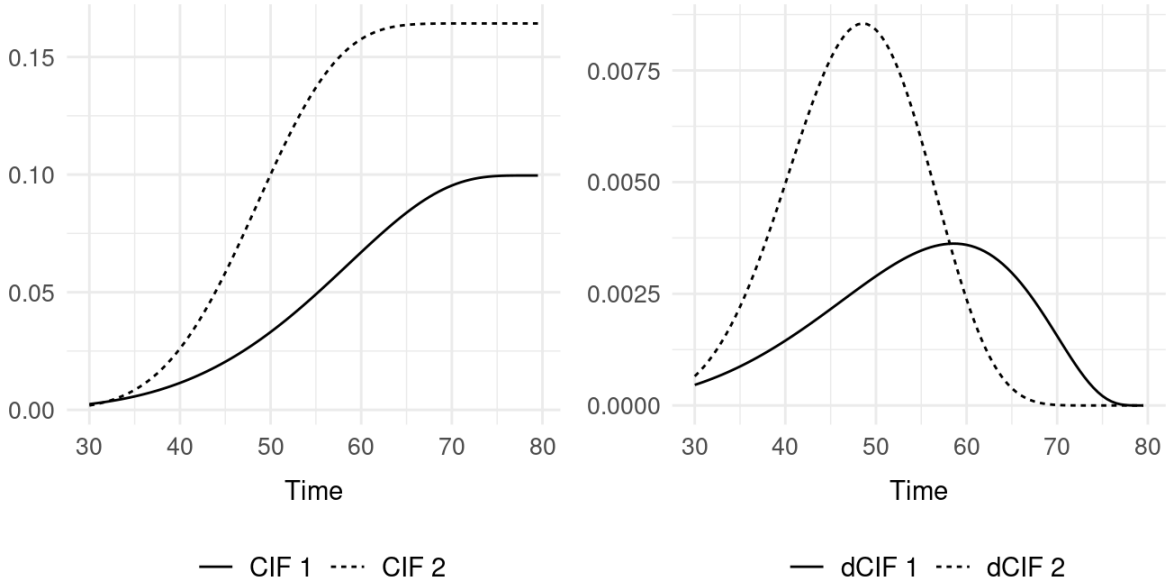
The model described in Equation 3.3 is in a general form, allowing for varying coefficients between clusters. However, we focus on a simpler structure with just fixed

intercepts. Fixing the latent effects in its distribution mean, zero, and using the following fixed effects configuration for two competing causes of failure

$$\begin{aligned} \boldsymbol{\beta} &= [-2 \quad 1.5]^\top \\ \boldsymbol{\gamma} &= [1.2 \quad 1]^\top \\ \boldsymbol{w} &= [3 \quad 5]^\top, \end{aligned} \tag{4.1}$$

we get the CIF's and failure probabilities (CIF derivatives w.r.t. time t, dCIF) presented respectively in Figure 9.

FIGURE 9 – CUMULATIVE INCIDENCE FUNCTIONS (CIF) AND RESPECTIVE DERIVATIVES (dCIF) W.R.T. TIME FOR A MODEL WITH TWO COMPETING CAUSES OF FAILURE, WITHOUT COVARIATES, LATENT EFFECTS AT ZERO, AND FIXED EFFECTS AS IN Equation 4.1



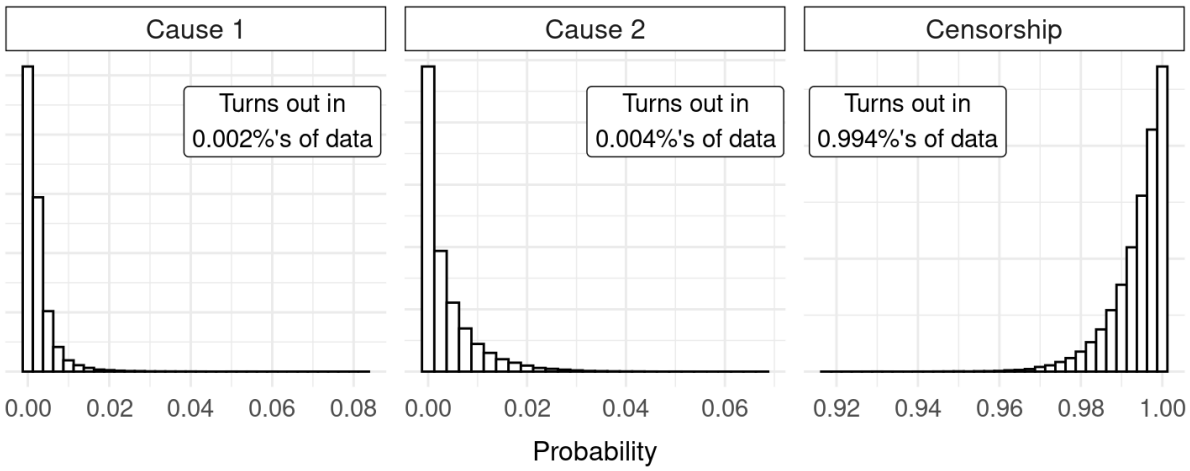
SOURCE: The author (2021).

By adding a complete latent structure,

$$\begin{bmatrix} u_1 \\ u_2 \\ \eta_1 \\ \eta_2 \end{bmatrix} \sim \text{Multivariate Normal} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.4 & 0.5 & 0.4 \\ & 1 & 0.4 & 0.3 \\ & & 1 & 0.4 \\ & & & 1 \end{bmatrix} \right), \tag{4.2}$$

we are able to apply Algorithm 1 and generate a complete model sample with 50000 clusters of size two (pairs), summarized in Figure 10. As already expected from Figure 9, where the CIF curves maximum are around 15%, the simulated failure probabilities and consequently the failure occurrences, are very small. In this way, we have a data sample filled with censorship, reflecting the reality of the applications. As disease incidences, for instance.

FIGURE 10 – SIMULATED FAILURE CAUSE PROBABILITIES WITH RESPECTIVE OUTPUT PCERCENTAGES FOR A MODEL WITH TWO COMPETING CAUSES AND 50000 CLUSTERS OF SIZE TWO. THE SIMULATION FOLLOWED ALGORITHM 1 GUIDELINES WITH PARAMETER CONFIGURATIONS SPECIFIED IN Equation 4.1 AND Equation 4.2



SOURCE: The author (2021).

The R function written to simulate the data is available in Appendix C.

In the simulation routine described in this section,the failure/censorship times are based on the repetition of an equally-spaced grid between 30 and 80-time units. A different approach but still non-parametric would be the random sampling of values between those limits. Yet, Cederkvist et al. (2019) does something different through the sampling of the censorship times from a $\mathrm{U}(0,\ \delta)$, and the sampling of $\varsigma \sim \mathrm{U}(0,\ 1)$ and the computation of the cause-specific failure times by solving

$$\varsigma = \Phi\left(w_k \text{arctanh}\left(\frac{t_{ij} - \delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}\right) \quad \text{for} \quad t_{ij},$$

with i being the subject, j the cluster, and k the failure cause.

This approach implies a parametric form for the failure times, which we do not know if holds in the real world.

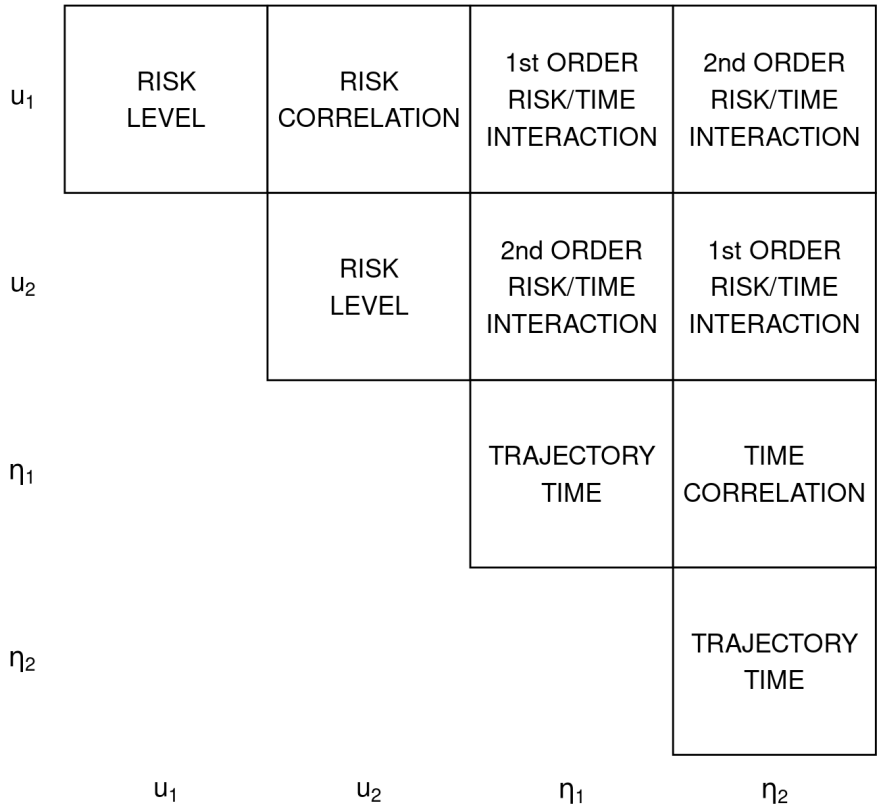
4.2 REAL-BASED DATASET

5 RESULTS

This chapter presents a simulation study results and the analysis of a real-based dataset.

5.1 SIMULATION STUDY

FIGURE 11 – BUILDING Σ



SOURCE: The author (2021).

Table 1 and Table 2

TABLE 1 – MODELS, FIRST PART

Label	Risk level		Trajectory time		Risk corr	Time corr	1st order risk/time interaction	2nd order risk/time interaction	Number of parameters
	$=$	$\neq$	$=$	$\neq$					
model1	✓								7
model2		✓							8
model3	✓				✓				8
model4		✓			✓				9
model5			✓						7
model6				✓					8
model7			✓			✓			8
model8				✓		✓			9
model9	✓		✓						8
model10		✓		✓					10
model11	✓		✓		✓				9
model12		✓		✓	✓				11
model13	✓		✓			✓			9
model14		✓		✓		✓			11
model15	✓		✓		✓	✓			10
model16		✓		✓	✓	✓			12
model17	✓		✓		✓	✓	✓		11
model18		✓		✓	✓	✓	✓		13
model19	✓		✓		✓	✓	✓		11
model20		✓		✓	✓	✓	✓		13

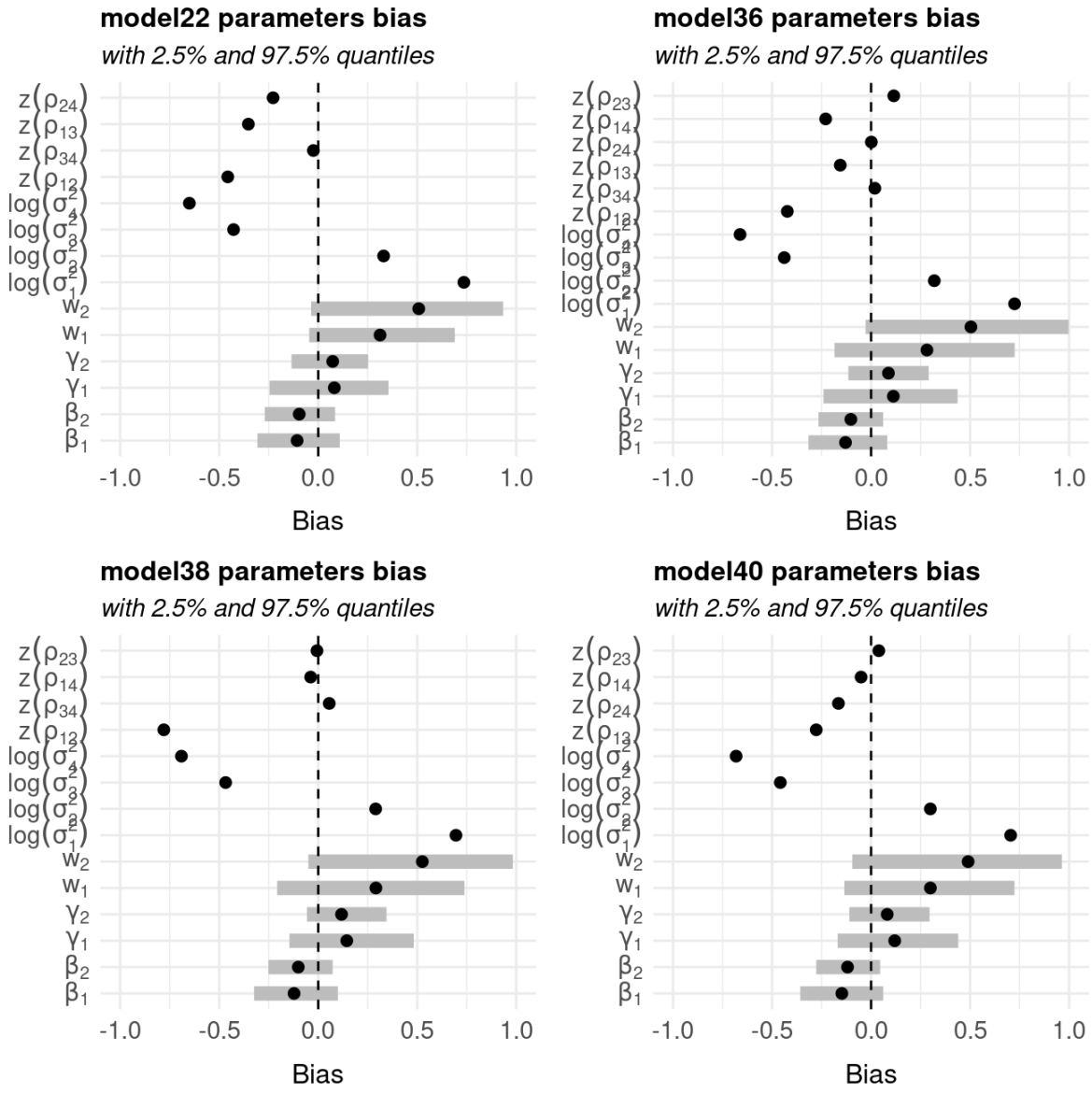
SOURCE: The author (2021).

TABLE 2 – MODELS, SECOND AND LAST PART

Label	Risk level		Trajectory time		Risk corr	Time corr	1st order risk/time interaction	2nd order risk/time interaction	Number of parameters
	$=$	$\neq$	$=$	$\neq$					
model21	✓		✓		✓	✓	✓✓		12
model22		✓		✓	✓	✓	✓✓		14
model23	✓		✓				✓		9
model24		✓		✓			✓		11
model25	✓		✓		✓		✓		10
model26		✓		✓	✓		✓		12
model27	✓		✓			✓	✓		10
model28		✓		✓		✓	✓		12
model29	✓		✓				✓✓		10
model30		✓		✓			✓✓		12
model31	✓		✓		✓		✓✓		11
model32		✓		✓	✓		✓✓		13
model33	✓		✓			✓	✓✓		11
model34		✓		✓		✓	✓✓		13
model35	✓		✓		✓	✓	✓✓	✓✓	14
model36		✓		✓	✓	✓	✓✓	✓✓	16
model37	✓		✓		✓	✓		✓✓	12
model38		✓		✓	✓	✓		✓✓	14
model39	✓		✓				✓✓	✓✓	12
model40		✓		✓			✓✓	✓✓	14

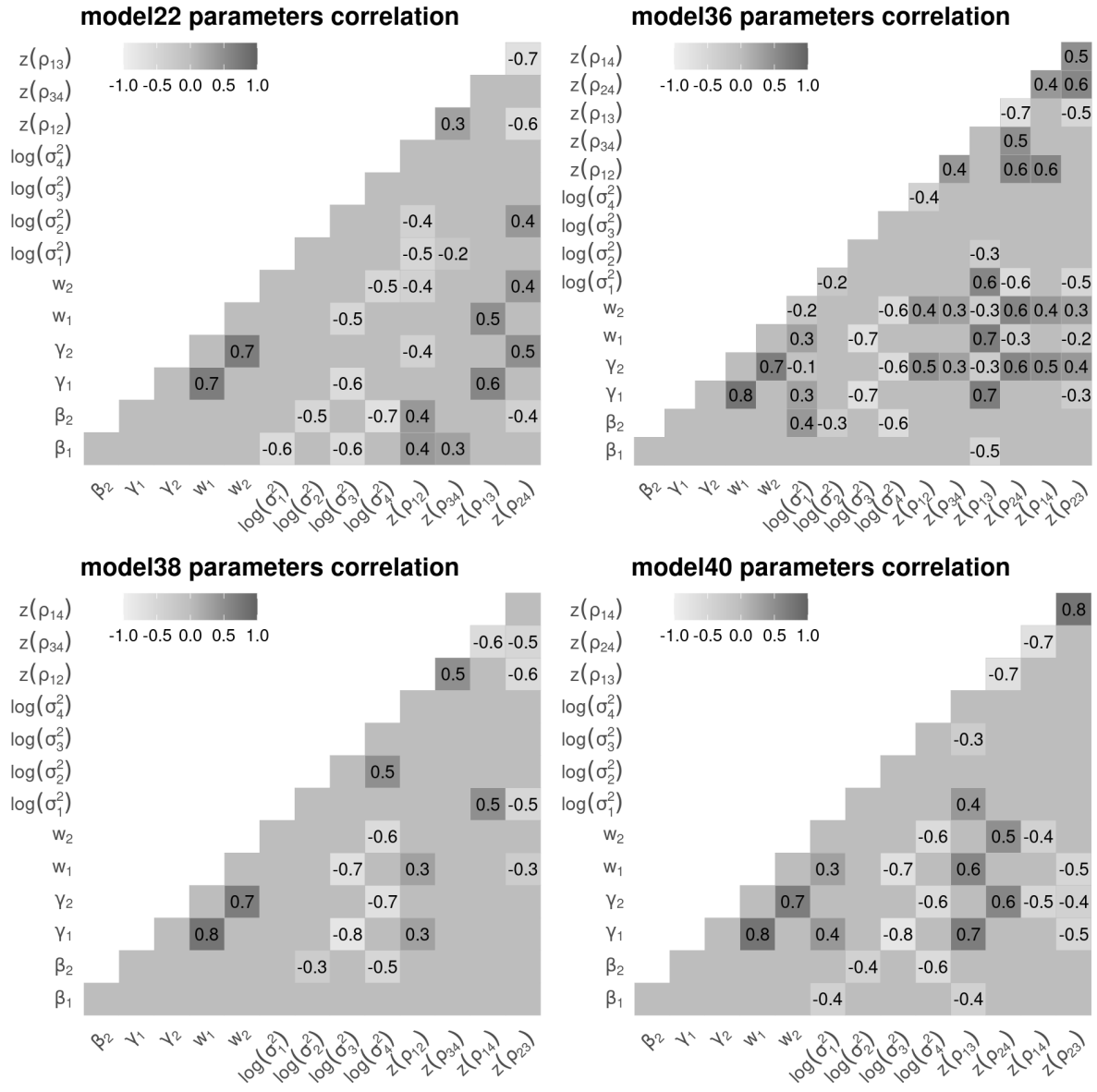
SOURCE: The author (2021).

FIGURE 12 – PARAMETERS BIAS WITH 2.5% AND 97.5% QUANTILES



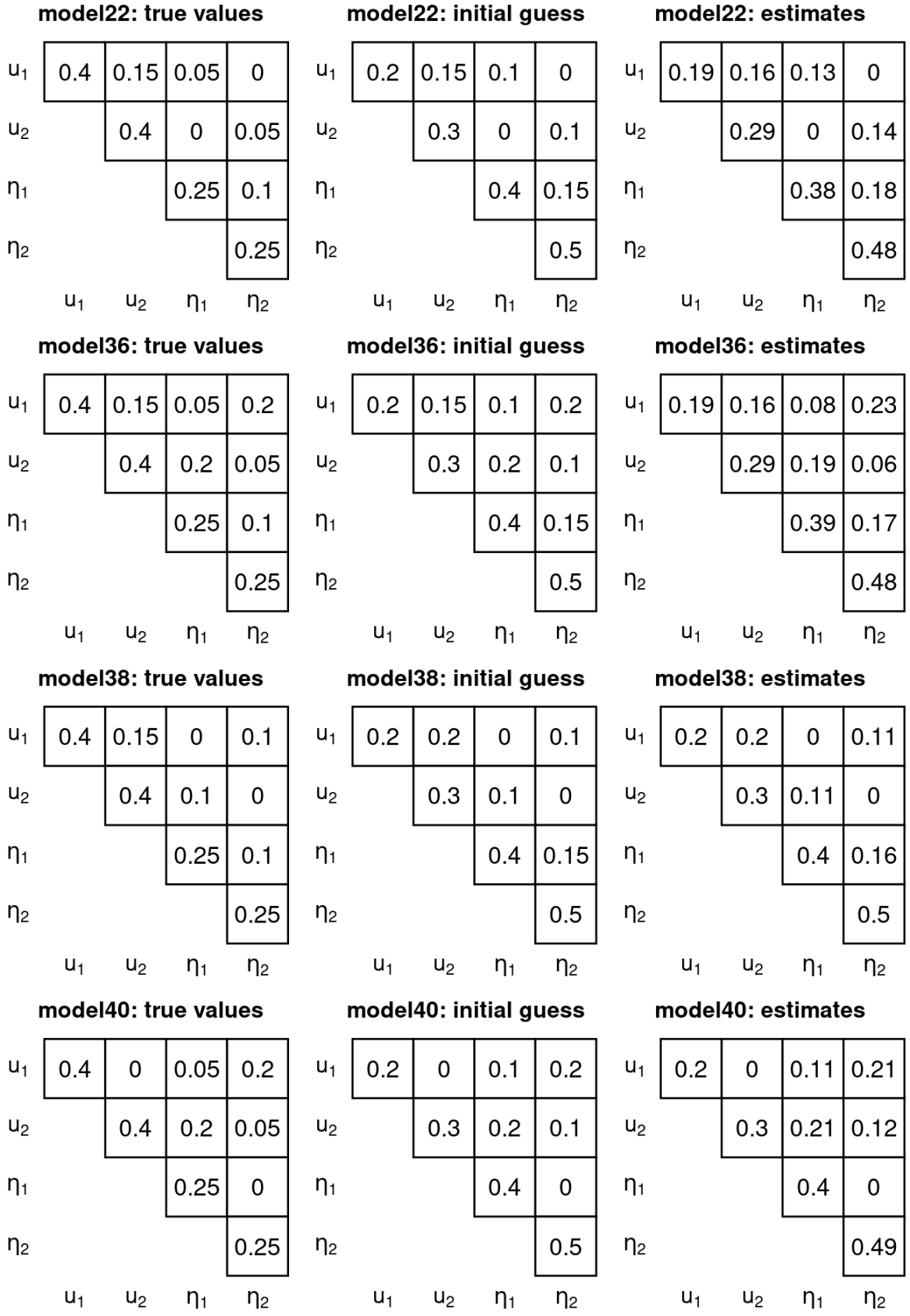
SOURCE: The author (2021).

FIGURE 13 – PARAMETERS CORRELATION



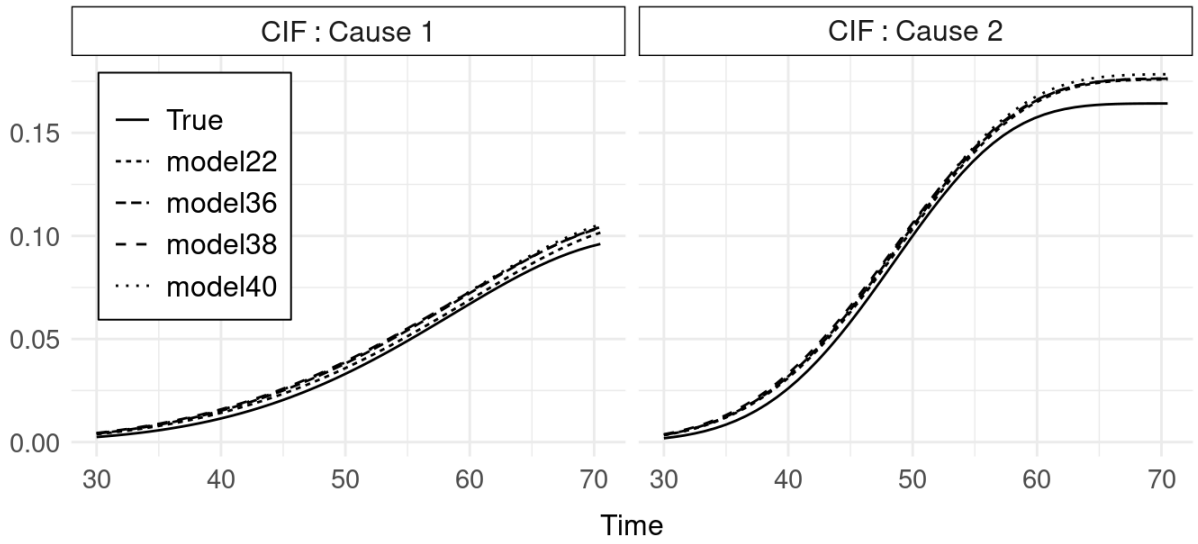
SOURCE: The author (2021).

FIGURE 14 – VARIANCE-COVARIANCE MATRIX UPPER-TRIANGULAR COMPONENTS



SOURCE: The author (2021).

FIGURE 15 – CUMULATIVE INCIDENCE FUNCTIONS (CIFs)



SOURCE: The author (2021).

5.2 REAL-BASED DATASET

6 FINAL CONSIDERATIONS

6.1 FUTURE WORKS

BIBLIOGRAPHY

ANDERSEN, P. K.; GESKUS, R. B.; WITTE, T. de; PUTTER, H. Competing risks in epidemiology: possibilities and pitfalls. *International Journal of Epidemiology*, v. 31, n. 1, p. 861–870, 2012. Cited on page 16.

BATES, D.; MAECHLER, M. *Matrix: Sparse and Dense Matrix Classes and Methods*. R Foundation for Statistical Computing. Vienna, Austria, 2019. R package version 1.2-18 ⟨https://CRAN.R-project.org/package=Matrix⟩. Cited on page 31.

BONAT, W. H.; RIBEIRO, P. J. Practical likelihood analysis for spatial generalized linear mixed models. *Environmetrics*, v. 27, n. 1, p. 83–89, 2016. Cited 2 times on pages 22 and 24.

CEDERKVIST, L.; HOLST, K. K.; ANDERSEN, K. K.; SCHEIKE, T. H. Modeling the cumulative incidence function of multivariate competing risks data allowing for within-cluster dependence of risk and timing. *Biostatistics*, v. 20, n. 2, p. 199–217, 2019. Cited 9 times on pages 8, 14, 16, 18, 34, 35, 36, 37, and 45.

CLAYTON, D. G. A model for association in bivariate life tables and its application in epidemiological studies of familial rendency in chronic disease incidence. *Biometrika*, v. 65, n. 1, p. 141–151, 1978. Cited on page 16.

COX, D. R.; REID, N. A note on pseudolikelihood constructed from marginal densities. *Biometrika*, v. 91, n. 3, p. 729–737, 2004. Cited on page 16.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B (Methodological)*, v. 39, n. 1, p. 1–38, 1977. Cited on page 22.

DENNIS, J. E.; GAY, D. M.; WELSCH, R. E. An Adaptive Nonlinear Least-Squares Algorithm. *ACM Transactions on Mathematical Software*, v. 7, n. 3, p. 348–368, 1981. Cited on page 26.

DIACONIS, P. The Markov chain Monte Carlo revolution. *Bulletin (New Series) of the American Mathematical Society*, v. 46, n. 2, p. 179–205, 2009. Cited on page 21.

FOURNIER, D. A.; SKAUG, H. J.; ANCHETA, J.; IANELLI, J.; MAGNUSSON, A.; MAUNDER, M. N.; NIELSEN, A.; SIBERT, J. AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software*, v. 27, n. 2, p. 233–249, 2012. Cited on page 30.

GAY, D. M. *Usage summary for selected optimization routines*. Computing Science Technical Report 153, AT&T Bell Laboratories. Murray Hill, NJ, 1990. Cited on page 26.

GELFAND, A. E.; SMITH, A. F. M. Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association*, v. 85, n. 410, p. 398–409, 1990. Cited on page 21.

GUENNEBAUD, G.; JACOB, B. et al. *Eigen v3*. 2010. ⟨http://eigen.tuxfamily.org⟩. Cited on page 31.

KALBFLEISCH, J. D.; PRENTICE, R. L. *The Statistical Analysis of Failure Time Data*. Second Edition. Hoboken, New Jersey: John Wiley & Sons, Inc., 2002. Cited 3 times on pages 14, 15, and 34.

KRISTENSEN, K.; NIELSEN, A.; BERG, C. W.; SKAUG, H. J.; BELL, B. M. TMB: Automatic Differentiation and Laplace Approximation. *Journal of Statistical Software*, v. 70, n. 5, p. 1–21, 2016. Cited 5 times on pages 18, 20, 30, 32, and 42.

LINDSAY, B. G. Composite likelihood methods. *Comtemporary Mathematics*, v. 80, n. 1, p. 221–239, 1988. Cited on page 16.

MCCULLAGH, P.; NELDER, J. A. *Generalized linear models*. Second edition. London: Chapman & Hall, 1989. Cited on page 17.

MCCULLOCH, C. E.; SEARLE, S. R. *Generalized, Linear, and Mixed Models*. New York: John Wiley & Sons, Inc., 2001. Cited 2 times on pages 17 and 20.

MOLENBERGHS, G.; VERBEKE, G. *Models for Discrete Longitudinal Data*. New York: Springer, 2005. Cited on page 22.

NELDER, J. A.; WEDDERBURN, R. W. M. Generalized linear models. *Journal of the Royal Statistical Society, Series A*, v. 135, n. 3, p. 370–384, 1972. Cited on page 17.

NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. Second Edition. New York: Springer, 2006. (Springer Series in Operations Research and Financial Engineering). Cited 3 times on pages 25, 26, and 27.

PEYRé, G. *Course notes on Optimization for Machine Learning*. 2020. May 10, ⟨https://mathematical-tours.github.io/book-sources/optim-ml/OptimML.pdf⟩. CNRS & DMA, École Normale Supérieure. Cited 2 times on pages 27 and 28.

PINHEIRO, J. C.; BATES, D. M. Unconstrained parametrizations for variance-covariance matrices. *Statistics and Computing*, v. 6, n. 3, p. 289–296, 1996. Cited on page 41.

PINHEIRO, J. C.; CHAO, E. C. Efficient Laplacian and Adaptive Gaussian Quadrature Algorithms for Multilevel Generalized Linear Mixed Models. *Journal of Computational and Graphical Statistics*, v. 15, n. 1, p. 58–81, 2006. Cited on page 22.

POURAHMADI, M. Cholesky decompositions and estimation of a covariance matrix: orthogonality of variance-correlation parameters. *Biometrika*, v. 94, n. 4, p. 1006–1013, 2007. Cited on page 42.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2021. ⟨https://www.R-project.org/⟩. Cited 3 times on pages 20, 26, and 30.

SHUN, Z.; MCCULLAGH, P. Laplace approximation of high dimentional integrals. *Journal of the Royal Statistical Society, Series B (Methodological)*, v. 57, n. 4, p. 749–760, 1995. Cited on page 22.

TIERNEY, L.; KADANE, J. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, v. 81, n. 393, p. 82–86, 1986. Cited on page 22.

VALPEL, J. W.; MANTON, K. G.; STALLARD, E. The impact of heterogeneity in Individual Frailty on the Dynamics of Mortality. *Demography*, v. 16, n. 1, p. 439–454, 1979. Cited on page 16.

VARIN, C.; REID, N.; FIRTH, D. An overview of composite likelihood methods. *Statistica Sinica*, v. 21, n. 1, p. 5–42, 2011. Cited on page 16.

Ver HOEF, J. M. Who Invented the Delta Method? *The American Statistician*, v. 66, n. 2, p. 124–127, 2012. Cited on page 32.

WOOD, S. N. *Core Statistics*. IMS: Institute of Mathematical Statistics, Textbooks, 2015. Cited 3 times on pages 22, 23, and 27.

Appendix

APPENDIX A – LATENT EFFECTS ANALYTIC GRADIENT FOR THE JOINT LOG-LIKELIHOOD FUNCTION OF THE MULTINOMIAL GLMM FOR CLUSTERED COMPETING RISKS DATA

The following gradient components are computed by cluster, to be used e.g., in a Newton optimization.

Subject i at cluster j and for competing cause k

$$
\begin{aligned}
&\frac{\partial}{\partial u_{kj}} \log L(\boldsymbol{\theta} \mid \boldsymbol{y}_j, \boldsymbol{r}_j) = \\
&y_{kij} \frac{1 + \sum_{m \neq k}^{K-1} \exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} - \left(\sum_{m \neq k}^{K-1} y_{mij}\right) \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} - \\
&y_{Kij} \frac{1}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} \Bigg(\\
&\frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\} \left(1 + \sum_{m \neq k}^{K-1} \exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}\right)}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} \times \\
&\frac{w_k \frac{\delta}{2\delta t - 2t^2} \phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}]}{1 - w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}]} - \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} \times \\
&\frac{\sum_{m \neq k}^{K-1} w_m \frac{\delta}{2\delta t - 2t^2} \phi[w_m \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi} - \eta_{mj}] \exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}}{1 - w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}]} \Bigg) - \\
&\boldsymbol{e}_k^\top \boldsymbol{Q} \boldsymbol{r}_j,
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial}{\partial \eta_{kj}} \log L(\boldsymbol{\theta} \mid \boldsymbol{y}_j, \boldsymbol{r}_j) = \\
&y_{kij}(w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}) - \\
&y_{Kij} \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} \times \\
&\frac{w_k \frac{\delta}{2\delta t - 2t^2} (w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}) \phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}]}{1 - \sum_{n=1}^{K-1} \frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}]} - \\
&\boldsymbol{e}_k^\top \boldsymbol{Q} \boldsymbol{r}_j,
\end{aligned}
$$

with $\boldsymbol{e}_k^\top$ begin a vector with 1 at the k-th position and zero elsewhere.

APPENDIX B – LATENT EFFECTS ANALYTIC HESSIAN FOR THE JOINT LOG-LIKELIHOOD FUNCTION OF THE MULTINOMIAL GLMM FOR CLUSTERED COMPETING RISKS DATA

The following hessian components are computed by cluster, to be used e.g., in a Newton optimization.

Subject i at cluster j and for competing cause k

$$
\begin{aligned}
&\frac{\partial^2}{\partial u_{kj}^2}\log L(\boldsymbol{\theta}\mid \boldsymbol{y}_j,\boldsymbol{r}_j) = \\
&-\frac{\left(\sum_{k=1}^{K-1} y_{kij}\right)\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki}+u_{kj}\}\left(1+\sum_{m\neq k}^{K-1}\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}\right)}{\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)^2}+ \\
&\frac{y_{Kij}\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki}+u_{kj}\}}{1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}\times \\
&\frac{\sum_{m\neq k}^{K-1} w_m\frac{\delta}{2\delta t-2t^2}\phi[w_m\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi}-\eta_{mj}]\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}}{1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}(1-w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}])}- \\
&\frac{y_{Kij}w_k\frac{\delta}{2\delta t-2t^2}\phi[w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj}]}{1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}\times \\
&\frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki}+u_{kj}\}\left(1+\sum_{m\neq k}^{K-1}\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}\right)}{1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}(1-w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}])}- \\
&\frac{y_{Kij}\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki}+u_{kj}\}}{\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)^2}\Bigg(\\
&\frac{\sum_{m\neq k}^{K-1} w_m\frac{\delta}{2\delta t-2t^2}\phi[w_m\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi}-\eta_{mj}]\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}}{\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}(1-w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}])\right)^2}- \\
&\frac{w_k\frac{\delta}{2\delta t-2t^2}\phi[w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj}]\left(1+\sum_{m\neq k}^{K-1}\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}\right)}{\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}(1-w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}])\right)^2}\Bigg) \\
&\times\Bigg(\Big(1+ \\
&\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}(1-w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}])\Big)+ \\
&\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)\times \\
&(1-w_k\frac{\delta}{2\delta t-2t^2}\phi[w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj}])\Bigg)-\boldsymbol{e}_k^\top\boldsymbol{Q},
\end{aligned}
$$

$$\frac{\partial^2}{\partial \eta_{kj}^2} \log L(\boldsymbol{\theta} \mid \boldsymbol{y}_j, \boldsymbol{r}_j) =$$

$$-y_{kij} - y_{Kij} \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} \Bigg($$

$$w_k \frac{\delta}{2\delta t - 2t^2} \phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}] \times$$

$$\frac{\left(w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}\right)^2 - 1}{1 - \sum_{n=1}^{K-1} \frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}]} -$$

$$\frac{\left(w_k \frac{\delta}{2\delta t - 2t^2} (w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}) \phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}]\right)^2}{\left(1 - \sum_{n=1}^{K-1} \frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}]\right)^2}$$

$$\Bigg) - \boldsymbol{e}_k^\top \boldsymbol{Q},$$

$$\frac{\partial^2}{\partial u_{kj} u_{mj}} \log L(\boldsymbol{\theta} \mid \boldsymbol{y}_j, \boldsymbol{r}_j) =$$

$$\left(\sum_{k=1}^{K-1} y_{kij}\right) \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\} \exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}}{\left(1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}\right)^2} +$$

$$\frac{y_{Kij} \exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\} \exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} \Bigg($$

$$\frac{w_m \frac{\delta}{2\delta t - 2t^2} \phi[w_m \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi} - \eta_{mj}]}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}(1 - w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}])} -$$

$$\frac{w_k \frac{\delta}{2\delta t - 2t^2} \phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}]}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}(1 - w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}])} \Bigg) -$$

$$\frac{y_{Kij}}{\left(1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}\right)^2} \Bigg(\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\} \Bigg($$

$$\frac{\sum_{m \neq k}^{K-1} w_m \frac{\delta}{2\delta t - 2t^2} \phi[w_m \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi} - \eta_{mj}] \exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}}{\left(1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}(1 - w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}])\right)^2} -$$

$$\frac{w_k \frac{\delta}{2\delta t - 2t^2} \phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}] \left(1 + \sum_{m \neq k}^{K-1} \exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}\right)}{\left(1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}(1 - w_n \frac{\delta}{2\delta t - 2t^2} \phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}])\right)^2} \Bigg)$$

$$\Bigg) \times \Bigg(\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}\Big(1+$$
$$\sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}(1 - w_n \frac{\delta}{2\delta t - 2t^2}\phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}])\Big) +$$
$$\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}(1 - w_m \frac{\delta}{2\delta t - 2t^2}\phi[w_m \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi} - \eta_{mj}])\Big(1+$$
$$\sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}\Big)\Bigg) - \boldsymbol{e}_k^\top \boldsymbol{Q},$$

$$\frac{\partial^2}{\partial \eta_{kj}\eta_{mj}} \log L(\boldsymbol{\theta} \mid \boldsymbol{y}_j, \boldsymbol{r}_j) =$$
$$- y_{Kij} \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} \times$$
$$\frac{w_k \frac{\delta}{2\delta t - 2t^2}(w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj})\phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}]}{\left(1 - \sum_{n=1}^{K-1} \frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} w_n \frac{\delta}{2\delta t - 2t^2}\phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}]\right)^2} \times$$
$$\frac{\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi} + u_{mj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} w_m \frac{\delta}{2\delta t - 2t^2}(w_m \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi} - \eta_{mj}) \times$$
$$\phi[w_m \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi} - \eta_{mj}] - \boldsymbol{e}_k^\top \boldsymbol{Q},$$

$$\frac{\partial^2}{\partial \eta_{kj} u_{kj}} \log L(\boldsymbol{\theta} \mid \boldsymbol{y}_j, \boldsymbol{r}_j) =$$
$$y_{Kij} \frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} \times$$
$$\frac{w_k \frac{\delta}{2\delta t - 2t^2}(w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj})\phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj})]}{\left(1 - \sum_{n=1}^{K-1} \frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}}{1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}} w_n \frac{\delta}{2\delta t - 2t^2}\phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}]\right)^2} \times$$
$$\Bigg(\sum_{n \neq k}^{K-1} \frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}}{\left(1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}\right)^2} \times$$
$$w_n \frac{\delta}{2\delta t - 2t^2}\phi[w_n \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni} - \eta_{nj}] -$$
$$\frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}\left(\left(1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}\right) - \exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki} + u_{kj}\}\right)}{\left(1 + \sum_{n=1}^{K-1} \exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni} + u_{nj}\}\right)^2} \times$$
$$w_k \frac{\delta}{2\delta t - 2t^2}\phi[w_k \text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right) - \boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki} - \eta_{kj}]\Bigg) -$$

$$y_{Kij}\frac{\frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki}+u_{kj}\}\left(\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)-\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki}+u_{kj}\}\right)}{\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)^2}}{1-\sum_{n=1}^{K-1}\frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}{1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}]}\times$$
$$w_k\frac{\delta}{2\delta t-2t^2}(w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj})\times$$
$$\phi[w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj}]-\boldsymbol{e}_k^\top\boldsymbol{Q},$$

$$\frac{\partial^2}{\partial\eta_{kj}u_{mj}}\log L(\boldsymbol{\theta}\mid\boldsymbol{y}_j,\boldsymbol{r}_j)=$$
$$y_{Kij}\frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki}+u_{kj}\}\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}}{\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)^2}\times$$
$$\frac{w_k\frac{\delta}{2\delta t-2t^2}(w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj})\phi[w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj})]}{1-\sum_{n=1}^{K-1}\frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}{1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}]}+$$
$$y_{Kij}\frac{\exp\{\boldsymbol{x}_{kij}\boldsymbol{\beta}_{ki}+u_{kj}\}}{1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}\times$$
$$\frac{w_k\frac{\delta}{2\delta t-2t^2}(w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj})\phi[w_k\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{kij}\boldsymbol{\gamma}_{ki}-\eta_{kj})]}{\left(1-\sum_{n=1}^{K-1}\frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}{1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}}w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}]\right)^2}\times$$
$$\left(\sum_{n\neq m}^{K-1}\frac{\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}}{\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)^2}\times\right.$$
$$w_n\frac{\delta}{2\delta t-2t^2}\phi[w_n\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{nij}\boldsymbol{\gamma}_{ni}-\eta_{nj}]-$$
$$\frac{\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}\left(\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)-\exp\{\boldsymbol{x}_{mij}\boldsymbol{\beta}_{mi}+u_{mj}\}\right)}{\left(1+\sum_{n=1}^{K-1}\exp\{\boldsymbol{x}_{nij}\boldsymbol{\beta}_{ni}+u_{nj}\}\right)^2}\times$$
$$\left.w_m\frac{\delta}{2\delta t-2t^2}\phi[w_m\text{arctanh}\left(\frac{t-\delta/2}{\delta/2}\right)-\boldsymbol{x}_{mij}\boldsymbol{\gamma}_{mi}-\eta_{mj}]\right)-\boldsymbol{e}_k^\top\boldsymbol{Q},$$

with $\boldsymbol{e}_k^\top$ begin a vector with 1 at the k-th position and zero elsewhere.

APPENDIX C – R CODE TO SIMULATE FROM A multiGLMM WITH TWO COMPETING CAUSES AND CLUSTERS OF SIZE TWO. FOR MORE INFORMATION CHECK SECTION 4.1

```
## input ------------------------------
## [numeric] number of clusters, J
## [vector] failure/censorship times t
## [Matrix] latent effects design matrix Z
## [matrix] variance-covariance matrix S
## output -----------------------------
## a [tibble] indicating the cluster element (i), the cluster (j), the
## failure/censorship probabilities (p1:p3), and the multinomial
## outcomes (y1:y3)

datasimu <- function(J, t, Z, S,
                     delta=80,
                     beta=c(-2, -1.5), gamma=c(1.2, 1), w=c(3, 5)){
    out <- tibble(i=rep(seq(2), times=J),
                  j=rep(seq(J), each=2), p1=NA, p2=NA, p3=NA)
    K <- dim(S)[1]/2+1
    ladim <- 2*(K-1) ## latent effects dimension
    B <- mvtnorm::rmvnorm(J, mean=rep(0, ladim), sigma=S)
    R <- Z%*%B
    risk1 <- exp(beta[1]+R[, 1])
    risk2 <- exp(beta[2]+R[, 2])
    level <- 1+risk1+risk2
    out$p1 <- risk1/level*w[1]*delta/(2*t*(delta-t))*
        dnorm(w[1]*atanh(2*t/delta-1)-gamma[1]-R[, 3])
    out$p2 <- risk2/level*w[2]*delta/(2*t*(delta-t))*
        dnorm(w[2]*atanh(2*t/delta-1)-gamma[2]-R[, 4])
    out <- out%>%mutate(p3=1-p1-p2)
    y <- mc2d::rmultinomial(2*J, 1, prob=out%>%select(p1:p3))
    out <- out%>%
        bind_cols(as_tibble(y))%>%
        rename(y1=V1, y2=V2, y3=V3)
    return(out)
}
library(tidyverse)
J <- 50e3
t <- rep(seq(from=30, to=79.5, by=0.5), length.out=2*J)
Z <- Matrix::bdiag(replicate(J, rep(1, 2), simplify=FALSE))
S <- matrix(c(1.0, 0.4, 0.5, 0.4,
              0.4, 1.0, 0.4, 0.3,
              0.5, 0.4, 1.0, 0.4,
              0.4, 0.3, 0.4, 1.0), nrow=4, ncol=4)
dat <- datasimu(J=J, t=t, Z=Z, S=S)
```