Approximate Bayesian Inference for Semi-parametric Proportional Hazard Models

*Ziang Zhang*

*2019/09/11 (ISO standard date format!)*

1. **Survival Analysis Model:**

**1.1 Introduction to Survival Analysis:**

Survival analysis refers to situations in which the response variable of interest is the time until the occurrence of a particular event. Examples include time to death of patients with a specific kind of disease, time to failure of a lightbulb, add one more example.. Models for analyzing time to event data quantify the association between treatments or risk factors and the time to an event. For example, we may quantify any relationship between the lifetimes of patients with the types of medicine they are using, to conclude whether a certain type of medicine is associated with a change in the overall survival times of patients.

Let T be a continuous non-negative random variable representing the time to some event, defined over the interval [0*,∞*). Let the probability density function of T be denoted as *f*(*t*) and its cumulative distribution function be *F*(*t*). The survivor function *S*(*t*) of T can be defined as:

# Z ∞

*S*(*t*) = *P*(*T > t*) = *f*(*x*)*dx* (1)

*t*

Notice that *S*(*t*) is the probability of an observation to survive to time t, and therefore it is a monotone decreasing function with *S*(0) = 1 andlim*t→∞ S*(*x*) = .

The hazard function, denoted h(t), is defined as instantaneous rate of occurrence at a specific time *t* given that the event does not occur before *t:*, called the

*P*(*t ≤ T ≤ t* + *s|T ≥ t*) *f*(*t*) *∂*

|  |  |
| --- | --- |
| *h*(*t*) = lim = = *−* log[*S*(*t*)] *s→*0 *s S*(*t*) *∂t*  The cumulative hazard function *H*(*t*) will be defined by: | (2) |
| Z *t*  *H*(*t*) = *h*(*u*)*du* = *−*log[*S*(*t*)] | (3) |

0

Event times are often only partially observed. Right censoring occurs when it is only known that an event occurred after some fixed timepoint, such as patients in a medical study who are lost to follow up. Left truncation occurs when event times are only included in the dataset conditional on having surpassed some threshold, such as patients joining a medical study only after having a disease for several weeks or months. Interval censoring refers to the presence of both right censoring and left truncation in the same observed event times. Partially observed observations present challenges in the analysis of survival data, and are too common to be ignored. We expand on these challenges in Section 1.2.

**1.2 Types of Censoring and Truncation:**

In survival analysis, we are mainly dealing with the problem of right-censoring, interval-censoring and left truncation. Right-censoring is when an individual’s lifetime *Ti* is not exactly known because the individual is still alive when the study terminates at *Ci*, so we are only sure about that *Ti > Ci* but not sure what exactly *Ti* is. Interval-censoring on the other hand, rises when the survival time is only known to be in an interval

(*Li,Ui*), and the left truncation problem happens when some survival times are not recorded unless they are bigger than a specified start time *ttr*, so all the data with survival times less than *ttr* are missed.

In general, we use the term “censoring” to refer to the scenarios where some lifetimes are only known to exceed their cutting times *Ci*, but we do not know how long do they last exactly. On the other hand, the term “truncation” mostly refer to a data collected problem where only lifetimes greater than the start time *ttr* are collected and observed. So these terms should be used in different situations depending on what kind of survival data are we dealing with.

There are two types of right-censoring that appear most frequently in the context of survival analysis, which are Type-I and Type-II right-censoring.

Type-I right-censoring occurs when each individual’s censoring time *Ci* is fixed and known beforehand. That means when we collect a bunch of survival times, we know whether each survival time is right-censored and when is it censored exactly. In this case, we will be able to write our original data-set *{Ti,Ci* : *i* = 1*,...,n}* as *{ti,δi* : *i* = 1*,...,n}* where:

*ti* = min*{Ti,Ci}, δi* = *I*(*Ti ≤ Ci*) (4)

This is the most common type of right-censoring, and we will focus on this type of censoring for the rest of the passage.

Type-II right-censoring occurs when we only observed the r smallest survival times in our sample. So the survival times that we can observed will be like *t*(1) *< t*(2) *< ... < t*(*r*), and the other survival times will be censored so we don’t know the exact numbers. In this scenario, we have a censoring time *t*(*r*) that is itself random.

Lastly, independent random censoring happens when both the ith survival time *Ti* and the ith censoring time *Ci* are random variable that are independent.

**1.3 Cox Proportional Hazard Model:**

In most survival analysis study, we are interested in incorporating some covariates *X*˜ = *{X*1*,X*2*,...,Xp}* into the distribution of survival time *T*, and studying their effects on the survival time *T*. Therefore, we often need to use different kinds of models to specify the dependence of T on *X*˜, and among those models, the proportional hazard model introduced by Cox(1972) is the most popular choice.

Let *h*(*t|x*˜) denote the hazard function of *T* at time t for a subject with covariates *x*˜ = (*x*1*,x*2*,...,xp*). The Cox Proportional Hazard Model can be specified as follows:

*h*(*t|x*˜) = *h*0(*t*)exp(*β*1*x*1 + *...* + *βpxp*) (5)

where *h*0(*t*) is an arbitrary baseline hazard function that does only depend on time, and *βi*’s are the unknown parameters that we are interested in estimating. The reason that it is called a “proportional” hazard model is because for any two subjects, the ratio of their hazard function will be constant over time. This is a modelassumption, and should be checked in practice when adopting the Cox PH model.

Notice that the baseline hazard function is left to be arbitrary, which implies that the Cox Proportional Hazard Model will be a semi-parametric model. There are different ways to define the baseline hazard functions, and the piece-wise constant baseline hazard model will be the most convenient and popular choice. We will focus on this kind of model in the rest of this paper, and I will introduce it in details in the next section.

**1.4 Proportional Hazard Model with Piece-wise Constant Baseline Hazard:**

Firstly, we break the time axis into K intervals with endpoints 0 = *s*0 *< s*1 *< ... < sK <* max*{ti* : *i* =

1*,...,n}*, and assumes that the baseline hazard function is constant in each interval. i.e: *h*0(*t*) = *λk* for *t ∈* (*sk−*1*,sk*)*, k* = 1*,*2*,...,K* Let *ηik* = log(*λk*) + *β*1*xi*1 + *...βpxip*, the model that we will be focusing on will be the semi-parametric proportional hazard model, specified at below:

*h*(*ti*) = *h*0(*ti*)exp(*β*1*xi*1 + *...βpxip*)

|  |  |  |
| --- | --- | --- |
| = exp[log(*λk*) + *β*1*xi*1 + *...βpxip*] *ti ∈* (*sk−*1*,sk*] = exp(*ηik*)  Using this information, we can derive the likelihood for that single observation to be: | | (6) |
| *L* = *f*(*ti*)*δiS*(*ti*)(1*−δi*)  = *h*(*ti*)*δiS*(*ti*) |  |
| Z *ti*  = exp(*δiηik*) exp*− h*(*u*)*du*  0  *k−*1  = exp(*δiηik*) exp*−* X(*sj − sj−*1)exp(*ηij*) *−* (*ti − sk−*1)exp(*ηik*)  *j*=1  Therefore, the full-likelihood of the data-set will be: |  | (7) |
| *n*  *k*(*i*)*−*1  Y X |  | (8) |
| *L* = exp(*δiηik*(*i*))exp *−* (*sj − sj−*1)exp(*ηij*) *−* (*ti − sk*(*i*)*−*1)exp(*ηik*(*i*))  *i*=1 *j*=1 | |

*n k*(*i*)*−*1

= Yexp*δiηik*(*i*) *−* X (*sj − sj−*1)exp(*ηij*) *−* (*ti − sk*(*i*)*−*1)exp(*ηik*(*i*))

*i*=1 *j*=1

I emphasize the subscript for *k*(*i*) because each survival time will correspond to a different value of k, depending on which interval the survival time lies in.

By taking the logarithm, the log-likelihood function for the *ith* observation *ti ∈* (*sk−*1*,sk*] can be written as :

*l* = log[*f*(*ti*)*δiS*(*ti*)(1*−δi*)]

= log[*h*(*ti*)*δiS*(*ti*)]

|  |  |  |
| --- | --- | --- |
| *k−*1  = *δiηik −* (*ti − sk−*1)exp(*ηik*) *−* X[(*sj − sj−*1)exp(*ηij*)]  *j*=1  Similarly, the full log-likelihood can be derived as: |  | (9) |
| *n*  *k*(*i*)*−*1  X X |  | (10) |
| *l* = *δiηik −* (*ti − sk −*1)exp(*ηik* ) *−* [(*sj − sj−*1)exp(*ηij*)] | |

(*i*) (*i*) (*i*)

*i*=1 *j*=1

It can see from the above expression that by considering a piece-wise constant baseline hazard, we make the corresponding log-likelihood much easier to work with, since the integral R0*ti h*(*u*)*du* can be replaced by a sum.

1. **INLA’s Inference Methodology:**

**2.1 Data Augmentation Using Poisson Likelihood:**

Here the INLA algorithm cannot directly be applied, because if we look at the log-likelihood of a single survival time *{ti,δi}*, we can find that it depends on more than one *η*. To use INLA, we required a conditional independent latent field together with a sparse Hessian matrix for the log-likelihood. That means we need to make sure that for a single data point, the log-likelihood should be free of terms from latent field once we condition on one of the term from the latent field.

To solve this puzzle, we will utilize a data “augmentation” trick to transform the log-likelihood of a single data point into the form that INLA likes. Notice that if we are looking at a random variable *Xi* that follows a Poisson distribution with mean (*ti − sk−*1)exp(*ηik*), then the log-likelihood corresponding to a single data point *{Xi* = 0*}* will be:

*l* = log *PXi* = 0*|λ* = (*ti − sk−*1)exp(*ηik*)

(11)

|  |  |
| --- | --- |
| = 0 *×* ln[(*ti − sk−*1)exp(*ηik*)] *−* (*ti − sk−*1)exp(*ηik*) *−* ln(0!) = *−*(*ti − sk−*1)exp(*ηik*)  Similarly, when *Xi* = 1, the log-likelihood of this single data point is:    *l* = log *P*(*Xi* = 1*|λ* = (*ti − sk−*1)exp(*ηik*)) |  |
| = 1 *×* ln((*ti − sk−*1)exp(*ηik*)) *−* (*ti − sk−*1)exp(*ηik*) *−* ln(1!) | (12) |

= ln(*ti − sk−*1) + *ηik −* (*ti − sk−*1)exp(*ηik*)

*∝ ηik −* (*ti − sk−*1)exp(*ηik*)

Here we can basically ignore the term ln(*ti − sk−*1) as it does not depend on any term from the latent field. So when we later take derivative, this term will just disappear which means it won’t affect our C matrix.

We showed that the first two terms of the log-likelihood of a single data point *{ti,δi}* can be viewed as the log-likelihood of a single data point *Xi ∼* Poisson*λ* = (*ti − sk−*1)exp(*ηik*)being 0 when *δi* = 0 and being 1 when *δi* = 1.

Next step will be to figure out a similar way to deal with the last term in equation (3). Notice that for a

Poisson random variable *Yj* with mean (*sj − sj−*1)exp(*ηij*), the log-likelihood for observing it being 0 will be:

*l* = log *PYj* = 0*|λ* = (*sj − sj−*1)exp(*ηij*)

(13) = *−*(*sj − sj−*1)exp(*ηij*)

Similarly, if we gather a sample of *{Yi*1 = 0*,Yi*2 = 0*,...,Yik* = 0*}* where each *Yij ∼* Poisson*λ* = (*sj − sj−*1)exp(*ηij*)is independent of others, then the log-likelihood of this sample will simply be the sum of log-likelihood of each term due to independence, which sums to be P*kj*=1*−*1(*sj − sj−*1)exp(*ηij*),that is exactly what we want.

Putting these two pieces information together, which means if we have a sample being *{Xi* = *δi,Yi*1 = 0*,Yi*2 = 0*,...,Yik* = 0*}*, and all the terms in this sample being mutually independent, then the log-likelihood of this sample will just be the log-likelihood of the single data point *{ti,δi}*. Doing this for all the data points *{ti,δi|i* = 1*,...,n}*. We retrieve the original log-likelihood from the log-likelihood of a sample of P*ni*=1 *k*(*i*) number of independent, but non-identical Poisson random variables. In other words, we augment our original data-set *{ti,δi|i* = 1*,...,n}* into a huge data-set*{xi,yi*1*,yi*2*,...,yik*(*i*)*|i* = 1*,*2*,...n}*, where all the terms in this new data-set are mutually independent. This is the cure for our problem since the log-likelihood of each term from this new “augmented” data-set, will only depend on the latent field through one *η*.

**2.2 Derivation of the Negated Hessian Matrix:**

Here I will present how the Bayesian approximation can be carried out using an INLA-type of algorithm.

Firstly, to make the covariance matrix of the joint Gaussian latent field non-singular, and to simplify the Hessian matrix that we are going to derive later, we will assume that for each *ηij*, a normal random noise *ij* is added. We assume that *ij* being mutually independent across different i and j. In other words, we will write *ηij* = log(*λk*) + *β*1*xi*1 + *...βpxip* + Γ*i* + *ηij*, where Γ*i* is any random effect that we believe exists in the context of the study.

Then, the latent field can be denoted as:

*W*˜ = *η*11*,η*12*,...,η*1*k,η*2*k,...,ηnk,*Γ1*,...,*Γ*q,β*1*,...,βp,*log(*λ*1)*,...,*log(*λk*)*T* (14)

Besides assume that *W*˜ is a GMRF, we also assume that log(*λk*+1)*−*log(*λk*) follows *N*(0*,τ−*1), a RW1 model. So we will just use *θ*˜ to denote the hyper-parameter vector that determines the precision matrix of our latent field.

Now, let’s derive the negated Hessian matrix of the log-likelihood with respect to the latent field. To do that, let’s first consider the log-likelihood consider only one survival time *{ti,δi}* where *ti ∈* (*sk*(*i*)*−*1*,sk*(*i*)]. In this case, the log-likelihood for this data point will be:

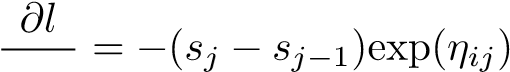
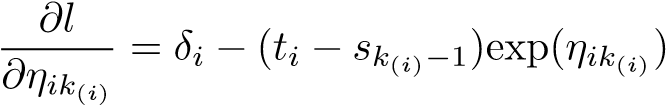
*k*(*i*)*−*1

*l* = *δiηik*(*i*) *−* (*ti − sk*(*i*)*−*1)exp(*ηik*(*i*)) *−* X [(*sj − sj−*1)exp(*ηij*)] (15)

*j*=1

The derivative with respect to *ηik*(*i*) will be

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *∂ηij*  (18) *∂*2*l*  *−*  = (*sj − sj−*1)exp(*ηij*)  *∂ηij*2  Apparently, for *ηij* where *j > k*(*i*), we have the second derivatives of log-likelihood being 0’s. Combine them | | | | | |  |  |
| together, we know that the negated Hessian matrix for the log-likelihood of *{ti,δi}*, *Hi* will be: | | | | |  |
| (*s*1 *− s*0)exp(*ηi*1)     0       0     ...       ...       ...       ...      0 | 0  (*s*2 *− s*1)exp(*ηi*2)  0  *···*  *···* | 0  0  ...  *···*  ...  *···* | *···*  ...  (*sk*(*i*)*−*1 *− sk*(*i*)*−*2)exp(*ηi*(*k*(*i*)*−*1))  0  ...  *···* | *···*  ...  ...  (*ti − sk*(*i*)*−*1)exp(*ηik*(*i*))  *···*  *···* | *···*  ...  *···*  ...  *···*  *···*  (19) | 0  ...  ...  ...  0 |                                      |

  (16)

|  |  |
| --- | --- |
| That means the negated second derivative will be: |  |
| *∂*2*l*  *−* 2 = (*ti − sk*(*i*)*−*1)exp(*ηik*(*i*))  *∂ηik*(*i*)  For first and negated second derivatives with *ηij* where *j < k*(*i*), we have: | (17) |

This is a very sparse matrix with only diagonal terms.

Repeating this procedure for the rest data points, using the property of independence, we can get the negated Hessian matrix H for the full log-likelihood will be:

*H*1 0 0 *···* 

 0 *H*2 0 *···*   ... 

 *···*



*H* =  ... (20)

 *···*

 *Hn*

 ... ...





0

Here we build a block diagonal matrix H using each block *Hi* obtained from above procedures. The negated Hessian matrix is very sparse, which is exactly what we want it to be. Then, we will try to derive the precision matrix Q of the latent field (To be continued).

1. **Proposed Methodology for Approximation:**

In the paper “Approximate Bayesian Inference for Case-Crossover Models”, the author suggested a new type of algorithm to do the approximation while allowing the log-likelihood of each observation to be dependent on more than one element from the latent field, which means the ad-hoc method using “data augmentation” is no longer needed (Stringer,2019). Here we will demonstrate how that algorithm can be used to estimate the parameters in Cox Proportional Hazard Model, and when this new algorithm will be preferred than INLA’s algorithm.

**3.1 Approximation using Partial Likelihood with Right censoring only:**

For simplicity, let’s assume that our main interest is the *βi*’s in the model but not the baseline hazard *h*0(*t*), and the only type of censoring present is right-censoring. Assume that *{ti* : *i* = 1*,...,k}* is a set of k distinct lifetimes that we actually *observed*, such that *t*(1) *< t*(2) *< ... < t*(*k*), and the result n-k lifetimes are the censored lifetimes that are not observed. Let *Ri* = *R*(*t*(*i*)) be the set of individuals who are alive and uncensored prior to time *t*(*i*) (including the i-th individual who dies at *t*(*i*)).

Define the hazard function for the i-th individual to be *h*0(*t*)exp(*ηi*), and let ∆*i,j* = *ηi − ηj*, then the partial likelihood for Cox Proportional Hazard Model can be written as:

Y*k* exp[*η*(*i*)]

*L*(*β*) =

P*l∈Ri* exp[*η*(*l*)] *i*=1

Y*k*  1

=

P*l∈Ri* exp[*η*(*l*) *− η*(*i*)]

*i*=1

(21)

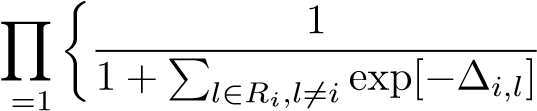
Y*k*  1

=

P*l∈Ri* exp[*−*∆*il*] *i*=1

*k*

=

*i*

Notice that this partial likelihood does not include any information on the baseline hazard function *h*0(*t*), meaning that all of the information are used to estimate the regression parameters in the model, which should result in a more precise estimation for them. Here it is that the partial likelihood only depend on those “differenced linear predictors” ∆*i,j*, so our latent field in this case will be *{*∆*,β,*Γ*}*. More importantly, because we are not estimating those baseline hazards, the algorithm’s convergence rate will be much faster. INLA does not allow this type of approximation because using partial likelihood to ignore the baseline hazard invalidates the “Poisson data-augmentation” trick that INLA does to make the C-matrix diagonal. While non-diagonal C matrix is not feasible in INLA’s algorithm, it will be feasible in the new proposed algorithm.

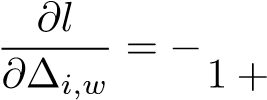
**3.1.1 Derivation of Hessian matrix and Precision matrix:**

For i-th observation, the partial log-likelihood will be:

*l* = *−*ln(1 + X exp[*−*∆*i,j*]) (22)

*j∈Ri,j6*=*i*

Therefore, taking derivative with respect to ∆*iw*, we can get:

exp(*−*∆*i,w*)

P*j∈R ,j6*=*i* exp[*−*∆*i,j*] (23)

*i*

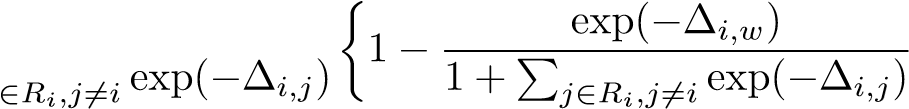
Similarly, we can see that:

*j Ri,j*

=

*∂*∆2 1 + P*j∈Ri,j6*=*i* exp(*−*∆*i,j*)2 *i,w*

(24)

 exp(*−*∆*i,w*)

=

1 + P*j*

Suppose that *M 6*= *w , M 6*= *i* and *M ∈ Ri*, then we also have:

*∂ l −*exp(*−*∆*i,w*) *∗ −* exp(*−*∆*i,M*)

= 2

*∂*∆*i,w*∆*i,M* 1 + P*j∈Ri,j6*=*i* exp(*−*∆*i,j*)

(25)



= 2

1 + P*j∈Ri,j6*=*i* exp(*−*∆*i,j*)

In this case, the latent field will be:

*W*˜ = (∆1*,*2*,*∆1*,*3*,...,*∆1*,n,*∆2*,*3*,*∆2*,*4*,...,*∆*n−*1*,n,...*)*T*

(26) = (∆˜1*,*∆˜2*,*∆˜3*,...*∆˜ *n−*1*,...*)*T*

where each ∆˜ *i* is defined as (∆*i,i*+1*,*∆*i,i*+2*,...,*∆*i,n*)*T*.

Using this notation, we can write the the negated hessian matrix *Ci* of the i-th observation’s log-likelihood, with respect to ∆˜ *i* being:

*− ∂*2*l − ∂*2*l − ∂*2*l ··· − ∂*2*l* 

*∂*∆2*i,i*+1 *∂*∆*i,i*+1∆*i,i*+2 *∂*∆*i,i*+1∆*i,i*+3 *∂*∆*i,i*+1∆*i,n*  *∂*∆*∂*2*i,i*2*l*+2 *−∂*∆*i,i*+2*∂*2∆*l i,i*+3 *··· −∂*∆*i,i∂*+22*l*∆*i,n* 

*−*



 *Ci* = 

 ...  (27)

 ... 



  

 

 *∂*2*l* 

*−∂*∆2*i,n*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Using these *Ci*’s as blocks, we can construct the *C* matrix for the full data-set being: | | | | | | |  |
| *C*1  0        *C* =           | 0  *C*2  *···* | 0  0  ... | *···*  *···*  ... | *Ck* | *···* |           ...  ...  0 | (28) |

The C matrix is block diagonal, but each block *Ci* is not diagonal. The reason for the C matrix to be block diagonal is that the log-partial-likelihood of i-th observation only depends on the latent field through the vector ∆˜ *i*, and for those ∆˜ *j* where *j > k*, they are not even included in the full log-partial-likelihood, so their corresponding *Cj* matrixes will be all zeroes. Though the C matrix is very sparse, but it is not a diagonal matrix, so INLA cannot handle this type of problem. However, our proposed algorithm could easily handle it, because diagonality of C matrix is not required here.

**3.2 Approximation using full-likelihood with left-truncation:**

If there are both right-censoring and left-truncations in our data-set, the “data augmentation trick” that INLA uses actually still works theoretically, but the software does not actually allow the user to run the approximation under this scenario. Fortunately, it can be solved using the package of the new proposed algorithm.

Recall that left-truncation happens when we cannot observed the i-th individual lifetime *ti*, unless it is greater than the entry time *ui*. Under this setup, all the observed lifetimes *ti*’s are known to be greater than their corresponding entry times *ui*’s. In other words, we should use conditional probability given *ti > ui* to form our likelihood. For simplicity, let’s still consider the same semi-parametric proportional hazard model with piece-wise constant basline hazard.

Denote the i-th lifetime as *ti*, the i-th left truncation time is *ui*, and assume that *ti ∈* (*sk*(*i*)*−*1*,sk*(*i*)], *ui ∈* (*sm*(*i*)*−*1*,sm*(*i*)]. Therefore, we have the likelihood being:

Y*n*  *f*(*ti*) *δi* *S*(*ti*) 1*−δi*

*L* =

*S*(*ui*) *S*(*ui*)

*i*=1

*n* 1*−δi*

*S*(*t*

= Y*f*(*ti*)*δi i*) (29)

*S*(*ui*)

*i*=1

*n*

= Y*h*(*ti*)*δi S*(*ti*)

*S*(*ui*)

*i*=1

Using the likelihood above, we can easily derive the log-likelihood being:

*n n*

*l* = X*δi*log[*h*(*ti*)] + Xlog*S*(*ti*) *−* log*S*(*ui*)(30)

*i*=1 *i*=1

Recall that if *ti ∈* (*sk*(*i*)*−*1*,sk*(*i*)], and *ui ∈* (*sm*(*i*)*−*1*,sm*(*i*)], we have the followings:

Z *ti*

log*S*(*ti*) = *− h*(*x*)*dx*

0

*k*(*i*)*−*1 (31)

= *−* X (*Sj − Sj−*1)exp(*ηij*) *−* (*ti − Sk*(*i*)*−*1)exp(*ηik*(*i*)) *j*=1

Similarly:

Z *ui*

log*S*(*ui*) = *− h*(*x*)*dx*

0

*m*(*i*)*−*1 (32)

= *−* X (*Sj − Sj−*1)exp(*ηij*) *−* (*ui − Sm*(*i*)*−*1)exp(*ηim*(*i*))

*j*=1

Therefore, the difference between this two terms can be written as:

Z *ti*

|  |  |
| --- | --- |
| log*S*(*ti*) *−* log*S*(*ui*) = *− h*(*x*)*dx*  *ui*  Whereas: | (33) |
| Z *ti k*(*i*)*−*1 X | (34) |
| *− h*(*x*)*dx* = *−* (*Sj − Sj−*1)exp(*ηij*) *−* (*Sm*(*i*) *− ui*)exp(*ηim*(*i*)) *−* (*ti − Sk*(*i*)*−*1)exp(*ηnik*(*i*)) |

*ui j*=*m*(*i*)+1

Then, combine all of the information above together, we can derive an expression for the log-likelihood of the sample:

*n n k*(*i*)*−*1 *n n*

*l* = X*δiηik*(*i*) *−* X X (*Sj − Sj−*1)exp(*ηij*) *−* X(*Sm*(*i*) *− ui*)exp(*ηim*(*i*)) *−* X(*ti − Sk*(*i*)*−*1)(*ηik*(*i*))

*i*=1 *i*=1 *j*=*m*(*i*)+1 *i*=1 *i*=1

(35)

If we have *m*(*i*) *≤ k*(*i*) *−* 1, then the above expression simplify to:

*n n n*

*l* = X*δiηik*(*i*) *−* X(*Sm*(*i*) *− ui*)exp(*ηim*(*i*)) *−* X(*ti − Sk*(*i*)*−*1)(*ηik*(*i*)) (36)

*i*=1 *i*=1 *i*=1

Next step, I will derive the corresponding C and Q matrix in this case.

**3.2.1 Derivation of C-matrix with left-truncation**

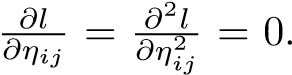
To make the derivation most general, I will assume that *k*(*i*) *−* 1 *≥ m*(*i*) + 1, since otherwise the computation will be simplified to trivial. For the i-th observation *t*(*i*) with left-truncation time *u*(*i*), assume that:

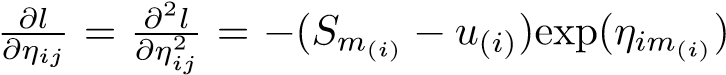
*ti ∈* (*sk*(*i*)*−*1*,sk*(*i*)], and *ui ∈* (*sm*(*i*)*−*1*,sm*(*i*)], then the likelihood of this observation will be:

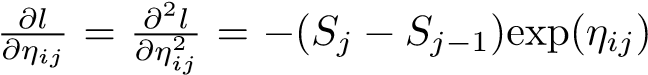
*k*(*i*)*−*1

*l* = *δiηik*(*i*) *−* X (*Sj − Sj−*1)exp(*ηij*) *−* (*Sm*(*i*) *− u*(*i*))exp(*ηim*(*i*)) *−* (*t*(*i*) *− Sk*(*i*)*−*1)exp(*ηik*(*i*)) (37)

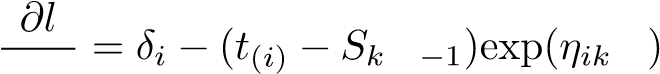
*m*(*i*)+1

For *j < m*(*i*) or *j > k*(*i*), apparently we have

For *j* = *m*(*i*), we can compute that .

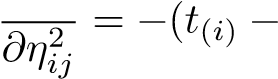
For *m*(*i*) *< j < k*(*i*), .

For *j* = *k*(*i*), it can be shown that:

(*i*) (*i*) (38)

*∂ηij*

So, *∂*2*l*

 *Sk*(*i*)*−*1)exp(*ηik*(*i*)) (39)

From now, let’s denote exp(*ηij*) as *bij*. Now, we can use the above information, to obtain the C-matrix (negated Hessian) of the i-th observation:

0 *··· ··· ···* 

0 ... 

... *···* (*Sm*(*i*) *− u*(*i*))*bim*(*i*) *··· ··· ···* 

... (*Sm*(*i*)+1 *− Sm*(*i*))*bi*(*m*(*i*)+1) *··· ···* 

 

... ... ... ... *···* 

 

... *···* (*Sk*(*i*)*−*1 *− Sk*(*i*)*−*2)*bi*(*k*(*i*)*−*1) *···* 

... *···* (*ti − Sk*(*i*)*−*1)*bik*(*i*) *···* 

0 *··· ··· ··· ··· ···* ... 

(40)

Let’s call the C-matrix of observation i as *Ci*, then the C-matrix of the whole sample will be:

*C*1 0 0 *···* 

 0 *C*2 0 *···*   ... 

*···*

*C* =  *Cn ···* ... (41)  ... ...



0

We can see that the present of left-truncation does not change the overall shape of the C-matrix. The only effect of it is to change the diagonal terms of each individual observation’s *Ci* matrix depending on the i-th left-truncation time. Therefore, the computation efficiency and precision will not be affected too much.

**3.3 Approximation using full-likelihood with interval-censoring:**

Suppose that we are not observing the exact lifetimes of individiuals, but only the set of intervals that contain each lifetime. In other words, our data-set is *{Li,Ri*;*i* = 1*,...n}* , where *Li ≤ ti ≤ Ri*. Using the same way to define the piece-wise constant hazard functions as before, we can assume that for the i-th observation, we have *Ri ∈* (*Sk*(*i*)*−*1*,Sk*(*i*)], and *Li ∈* (*Sm*(*i*)*−*1*,Sm*(*i*)].

Now, we can write down the log-likelihood of the i-th individual using the above information:

Z *Ri*  Z *Ri*

*li* = *δi*log*hi*(*Ri*)*− hi*(*u*)*du* + log 1 *−* exp*− hi*(*u*)*du*

0 *Li*

*k*(*i*)*−*1 (42)

= *δiηiki −* X exp(*ηij*) *−* (*Ri − Sk*(*i*)*−*1)exp(*ηik*(*i*)) + log1 *−* exp(*ϑi*)

*j*=1

Where *ϑi* is defined as:

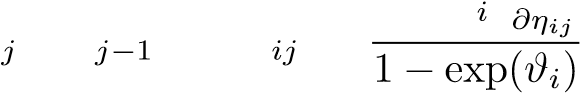
Z *Ri k*(*i*)*−*1

*ϑi* = *− hi*(*u*)*du* = *−* X (*Sj − Sj−*1)exp(*ηij*) *−* (*Sm*(*i*) *− Li*)exp(*ηim*(*i*)) *−* (*Ri − Sk*(*i*)*−*1)exp(*ηik*(*i*))

*Li j*=*m*(*i*)+1

(43)

Now, we can take derivative of this log-likelihood with respect to the ij-th linear predictor (assume that *m*(*i*) + 1 *≤ j ≤ k*(*i*) *−* 1), and get the following result:

 *∂li* = *−*(*S − S* )exp(*η* ) *−* exp(*ϑ* ) *∂ϑi* (44)

*∂ηij*

Where *∂η∂ϑiji* is *−*(*Sj − Sj−*1)exp(*ηij*) in this case.

Since *ϑi* depends on more than one linear predictors, so it follows naturally that *∂η∂liji* will be a function of several different linear predictors, which means the C-matrix of each observation’s log-likelihood will be non-diagonal. That will be a serious problem for INLA, as its package relies on the diagonality of C-matrix, but it will not cause any probelm for our new proposed algorithm, as the diagonality of C-matrix is no longer required here.

**4 Example from Diabetics Data-set**

Firstly, I will use the data-set “diabetics” to demonstrate the equivalence between “coxph” approach in survival package, INLA’s approach and our proposed approach. This data-set contains the results from a trail of laser coagulation for the treatment of diabetic retinopathy from 197 patients. Each patient had one eye randomized to laser treatment and the other eye received no treatment. (To be continued on “proposed approach”)

The variable “id” specifies the subject’s ID.

The variable “laser” is a categorical variable with levels xenon or argon.

The variable “age” is the age of the subject at diagnosis.

The variable “eye” is a categorical variable with levels left or right.

The variable “trt” is a categorical variable with levels 0 for no treatment and 1 for treatment using laser.

The variable “risk” classifies the risk levels of the patients.

The response variable in this data-set will be “time”, which are the actual time to blindness in months, minus the minimum possible time to event (6.5 months), and “status” indicates whether the time is censored with 1 for visual loss and 0 for censored. The censoring can be due to death, dropout, or end of the study.

Let use briefly view the structure of “diabetics”:

**4.1 Data-set:**

**head**(**as\_tibble**(diabetic))

## # A tibble: 6 x 8

## id laser age eye trt risk time status

## <int> <fct> <int> <fct> <int> <int> <dbl> <int>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ## 1 | 5 argon | 28 left | 0 | 9 46.2 | 0 |
| ## 2 | 5 argon | 28 right | 1 | 9 46.2 | 0 |
| ## 3 | 14 xenon | 12 left | 1 | 8 42.5 | 0 |
| ## 4 | 14 xenon | 12 right | 0 | 6 31.3 | 1 |
| ## 5 | 16 xenon | 9 left | 1 | 11 42.3 | 0 |
| ## 6 | 16 xenon | 9 right | 0 | 11 42.3 | 0 |

We can see that those survival times are right-censored. We will fit a cox proportional hazard model with piece-wise constant baseline hazard, assuming that each individual will have the same baseline hazard function. The variable ID will be treated as a random effect(which can be added using code “frailty.gaussian(id)”). The variables “age”, “eye”, “trt” and “laser” will be included as fixed effects.

**4.2 Survival: coxph**

|  |
| --- |
| diabetic.CoxPh <- **coxph**(**Surv**(time, status)**~**age **+** eye **+** trt **+** laser **+ frailty.gaussian**(id), **summary**(diabetic.CoxPh) |

data = diabet

## Call:

## coxph(formula = Surv(time, status) ~ age + eye + trt + laser +

## frailty.gaussian(id), data = diabetic)

##

## n= 394, number of events= 155

##

|  |  |
| --- | --- |
| ## | coef se(coef) se2 Chisq DF p |
| ## age | 0.009548 0.01323 0.009879 0.52 1.00 4.7e-01 |
| ## eyeright | 0.483005 0.17501 0.168693 7.62 1.00 5.8e-03 |
| ## trt | -1.007507 0.17930 0.174315 31.57 1.00 1.9e-08 |
| ## laserargon | -0.182388 0.39471 0.293566 0.21 1.00 6.4e-01 |

## frailty.gaussian(id) 131.35 79.63 2.4e-04

##

## exp(coef) exp(-coef) lower .95 upper .95

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ## age | 1.0096 | 0.9905 | 0.9837 | 1.0361 |
| ## eyeright | 1.6209 | 0.6169 | 1.1503 | 2.2842 |
| ## trt | 0.3651 | 2.7388 | 0.2569 | 0.5189 |
| ## laserargon | 0.8333 | 1.2001 | 0.3844 | 1.8062 |

##

## Iterations: 6 outer, 24 Newton-Raphson

## Variance of random effect= 0.9447455

## Degrees of freedom for terms= 0.6 0.9 0.9 0.6 79.6

## Concordance= 0.867 (se = 0.867 )

## Likelihood ratio test= 228 on 82.62 df, p=2e-15

From the output above, it can be seen that variables “age” and “eyeright” have positive association with the rate of occurrence of visual loss, and variables “trt” and whether using “argon” type of laser are negatively associated with the rate. All the fixed effects that we included in this study have significant effects for the risk of visual loss.

**4.3 Bayesian: INLA**

Now we fit the same model using INLA:

formula = **inla.surv**(time, status) **~** age **+** eye **+** trt **+** laser **+ f**(id, model = "iid") diabetic.INLA <- **inla**(formula, control.compute = **list**(dic = TRUE), family = "coxph", data = diabetic, control.hazard=**list**(model="rw2", n.intervals=20))

diabetic.INLA**$**summary.fixed

## mean sd 0.025quant 0.5quant 0.975quant

## (Intercept) -4.763729393 0.24996255 -5.27247610 -4.757707446 -4.28881333

## age 0.007160643 0.01085699 -0.01385412 0.007024607 0.02907964

## eyeright 0.385802068 0.17639593 0.04935632 0.382261555 0.74288124

## trt -0.867311186 0.19243839 -1.26426547 -0.860445271 -0.50841633

## laserargon -0.118830760 0.32297942 -0.77672824 -0.111820255 0.49864364

## mode kld

## (Intercept) -4.746079947 3.457442e-06

## age 0.006834706 8.068802e-06

## eyeright 0.375286188 5.876295e-06

## trt -0.846080656 1.415197e-05

## laserargon -0.099026290 3.739273e-06

It seems like these two approaches are similar enough. Though in the classic “coxph” approach, the effects of age and using argon-type laser are significant, but INLA gives insignificant results(the 95% credible interval contains zero). There is an estimate for intercept in the INLA’s method because we used a random walk prior in that.

1. **Example from Bladder Data-set:**

Next, we will study the two approaches on the data-set “bladder1”. This is the full data-set that contains the result from a study on recurrences of bladder cancer from 118 subjects. In this data-set, the variables that we are interested in are “id”, “number”, “size”, “recur”, “times” and “censored”.

The variable “id” is the patient ID.

The variable “number” specifies initial number of tumors of each subject.

The variable “size” is the size of largest initial tumor.

The variable “recur” is the number of recurrence of bladder cancer for that subject.

The response variable will be “time” which is computed to be the duration of times until recurrence or death, censored by the variable “censored” with 0 means being censored.

**5.1 Data-set:**

|  |
| --- |
| data <- **as\_tibble**(bladder1) data <- **select**(data,**-c**(rsize,rtumor,enum)) data <- data **%>% mutate**(censored = status**==**0) **for** (i **in** 1**:length**(data**$**censored)) { **if**(data**$**censored[i]) data**$**censored[i] <- 0 **else** data**$**censored[i] <- 1  } data <- data **%>% mutate**(times = stop**-**start) **head**(data) |

## # A tibble: 6 x 10

## id treatment number size recur start stop status censored times

## <int> <fct> <int> <int> <int> <int> <int> <dbl> <dbl> <int>

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ## 1 1 | placebo | 1 | 1 | 0 | 0 | 0 | 3 | 1 | 0 |
| ## 2 2 | placebo | 1 | 3 | 0 | 0 | 1 | 3 | 1 | 1 |
| ## 3 3 | placebo | 2 | 1 | 0 | 0 | 4 | 0 | 0 | 4 |
| ## 4 4 | placebo | 1 | 1 | 0 | 0 | 7 | 0 | 0 | 7 |
| ## 5 5 | placebo | 5 | 1 | 0 | 0 | 10 | 3 | 1 | 10 |
| ## 6 6 | placebo | 4 | 1 | 1 | 0 | 6 | 1 | 1 | 6 |



Here the variable “id” specifies different individuals, and should be treated as a random effect. The variable

“time” is computed using the difference between variables “start” and “stop”, which denote the start time and end time of each time interval. It seems like a interval censoring problem but the start time is known before hand, so we can treat it as a regular type-I right censoring.

In this study, a nonzero value of “status” can be death from bladder disease, death from other reason or recurrence. Here we will just view all of these situations as “occurrence” for simplicity. So the variable

“censored” is created such that it is 1 if “status” is non-zero, otherwise 0. We will include “number”, “size” and “recur” as fixed effects in this study.

**5.2 Survival:coxph**

|  |
| --- |
| bladder.CoxPh <- **coxph**(**Surv**(times, censored)**~** number **+** size **+** recur **+ frailty.gaussian**( **summary**(bladder.CoxPh) |

id), data = data)

## Call:

## coxph(formula = Surv(times, censored) ~ number + size + recur +

## frailty.gaussian(id), data = data)

##

## n= 294, number of events= 218

##

## coef se(coef) se2 Chisq DF p

|  |  |
| --- | --- |
| ## number | 0.064848 0.04246 0.04051 2.33 1.00 1.3e-01 |
| ## size | 0.008336 0.04797 0.04631 0.03 1.00 8.6e-01 |
| ## recur | 0.229848 0.02559 0.02431 80.66 1.00 2.7e-19 |

## frailty.gaussian(id) 5.12 4.59 3.5e-01

##

## exp(coef) exp(-coef) lower .95 upper .95

## number 1.067 0.9372 0.9818 1.160

## size 1.008 0.9917 0.9179 1.108

## recur 1.258 0.7947 1.1968 1.323

##

## Iterations: 8 outer, 43 Newton-Raphson

## Variance of random effect= 0.0257231

## Degrees of freedom for terms= 0.9 0.9 0.9 4.6

## Concordance= 0.694 (se = 0.694 )

## Likelihood ratio test= 105.4 on 7.33 df, p=<2e-16

Fitting this model using the traditional partial likelihood approach gives insignificant results for all the fixed effects except “recur”, which has a strong positive effect. But we will still proceed to check what will happen if we fit it using a Bayesian approach.

**5.3 Bayesian: INLA**

formula = **inla.surv**(times, censored) **~** -1 **+** number **+** size **+** recur **+ f**(id, model = "iid") bladder.INLA <- **inla**(formula, control.compute = **list**(dic = TRUE), family = "coxph", data = data, control.hazard=**list**(model="rw1", n.intervals=20))

bladder.INLA**$**summary.fixed

## mean sd 0.025quant 0.5quant 0.975quant mode

## number -0.4033390 0.11182193 -0.6273874 -0.4019897 -0.1869181 -0.3992881

## size -0.7277341 0.11391532 -0.9608482 -0.7245355 -0.5126344 -0.7182927

## recur 0.3002297 0.09091334 0.1254882 0.2987734 0.4835221 0.2959428

## kld

## number 3.515352e-05

## size 6.173690e-05

## recur 7.080553e-06

Indeed, the two results seem pretty similar in general. In both cases, we can see that there are no apparent relationships between all of the fixed effects and the rate of occurrence of bladder cancer’s recurrence, or death, except the variable “recur” with a positive effect.