# Joint Modelling of Longitudinal Mixed Effects and Accelerated Failure Time Model

*by*


**Koushik Khan**

Reg. No.: VB-299 of 2010-11
Roll No.: MSC(Sem-IV)-Stat-223


*A dissertation presented for the degree of*
*Master of Science*
*in Statistics*

***supervised by***
Dr. Arindom Chakraborty
Assistant Professor



Department of Statistics
Visva-Bharati
Santiniketan, India

22nd June, 2015

# Acknowledgement

At the very beginning I do like to acknowledge the persons who helped me in various ways in completing this project. At the very inception, I would like to express sincere gratitude to my respected teacher cum project supervisor Dr. Arindom Chakraborty, Department of Statistics, Visva-Bharati, for his continuous co-operation and valuable suggestions throughout the period of this project work, failing of which it would not have been possible for me to complete my work.

I would like to give special thanks to my father Mr. Keshab Khan, my mother Smt. Pratima Khan, Mr. Joy Mustafi (Joy da) and Mrs. Tanusree Mustafi (Mohor di). Joy da and Mohor di guided me with special care to take Statistics as a subject for my undergraduate course at the end of my +2 level.

My deep regard, respect as well as debt goes to all my respected teachers of Department of Statistics, Visva-Bharati for their valuable suggestions and ideas. I am also thankful to all of my classmates, specially Anish, Mahaprasad, Shovan and Supriyo, who helped me in many situations by giving theoretical and programming ideas.

Finally, I wish to dedicate my master's thesis to my teacher Late **Sri Samatul Krishna Roy**, without his care at my 10th level, I might not be present here.

*Koushik Khan*
*Department of Statistics*
*Visva-Bharati*
*22nd June, 2015*

# Abstract

In clinical trials and other follow-up studies, it is natural that a response variable is repeatedly measured during follow-up and the occurrence of some key event is also monitored. There has been a considerable study on the joint modelling of these measures together with information on covariates. The joint modelling of such serial outcomes and the time-to-event data looks a bit complicated. Computational complexities arise due to the presence of subject specific random effects. The primary objective of this project is to obtain the estimates of the parameters. In the absence of any closed form solution for the estimating equations, a computational intensive method is proposed. Several sets of simulation studies are performed using the R Software (http://cran.r-project.org/) to study the performance of the proposed Monte Carlo EM (MCEM) technique under small sample.

This project is actually motivated by a data set concerning longitudinal outcomes of subjects involved in a study on muscular dystrophy syndrome among the children caused by deletion, duplication or point mutation of the Dystrophin gene located on X-chromosome. These values consist of observations on ten different muscles that are responsible for walking. Two time-to-event indicators are also observed over different time points. These are time taken by a patient to walk 4 steps and to get up from lying state. Censoring occurs if a person fails to complete the four steps in 1 min or fails to stand up within 40 s. As both the causes are highly dependent, there is no harm in assuming that both the failures occur simultaneously. These scores, observed up to failure from any of the causes for each subject, are ordinal in nature. Our interest is on characterizing the relationship between failure time (due to any one of the causes) and the longitudinal outcomes. In this study only the first one (complete the four steps in 1 min) is used, which appears to be more severe according to the doctors.

# Contents

# 1 Introduction

Repeated measurement studies, in particular, longitudinal studies, are important tools in epidemiological, clinical and social science research. Repeated measure models usually include an underlying "'functional"' relationship between at least one of the predictor variables and observations within individuals. A common type of repeated measure data is longitudinal data where the observations are ordered by time or position in space.

Longitudinal models with linear relationship between outcomes and covariates are quite popular in practice. In such studies, measurements of a subject at one time point may depend on measurement on previous time points. The observations corresponding to each individual (cluster) may be correlated. Different subjects, however, are assumed to be independent. Ignoring this correlation between measurements of a subject in the context of regression model may be fatal.

Mixed effects models for repeated measures data have become popular because of their flexible covariance structure. Such models allow for non-constant correlation among the observations. Moreover, in mixed effects models it is assumed that individuals' response follow a functional form with parameters varying among individuals.

In practice, longitudinal data and survival data frequently arise together. For example, in many medical studies, we often collect patients' information (e.g., blood pressures) repeatedly over time and we are also interested in the time to recovery or recurrence of a disease. Longitudinal data and survival data are often associated in some ways. The time to event may be associated with the longitudinal trajectories. Separate analyses of longitudinal data and survival data may lead to inefficient inference. Joint models of longitudinal and survival data, on the other hand, incorporate all information simultaneously and provide valid and efficient inferences.

A typical model setting is to assume a mixed-effects model for the longitudinal data and a Cox model or an accelerated failure time (AFT) model for the survival data, with the two models sharing some random effects or variables. The likelihood method is often used, implemented by EM algorithms.
In the context of joint modeling it is necessary to establish a clear framework to distinguish terminology from longitudinal and time-to-event processes. We are interested in two processes, the longitudinal $\boldsymbol{Y}$ and time-to-event $\boldsymbol{T}$. When the event is not observed, the censoring ($\boldsymbol{C}$) occurs.

There are two strategies to factorize the joint density of ($\boldsymbol{Y}$, $\boldsymbol{T}$) (Little(1995)) based on model interpretations, and consequently suitability for individual problems. These are *Selection model* and *Pattern mixture model* which factorizes [$\boldsymbol{Y}$, $\boldsymbol{T}$] as

$$\begin{array}{cc} \textit{Selection model} & \textit{Pattern mixture model} \\ [\boldsymbol{Y}, \boldsymbol{T}] = [\boldsymbol{T}|\boldsymbol{Y}][\boldsymbol{Y}] & [\boldsymbol{Y}, \boldsymbol{T}] = [\boldsymbol{Y}|\boldsymbol{T}][\boldsymbol{T}] \end{array}$$

Mathematically speaking, both models describe exactly the same joint distribution, but their statistical interpretations are different. When it is of interest to have inference on time to event parameters, selection model is used. On the contrary, when primary interest is on the longitudinal trajectory, which might be associated with an event pattern, pattern mixture model is commonly used. This implies that, the two approaches lead to different understanding and inferences on the model parameters.

Both selection model and pattern mixture model can be extended to incorporate random effects resulting in random selection model and random pattern mixture model respectively.

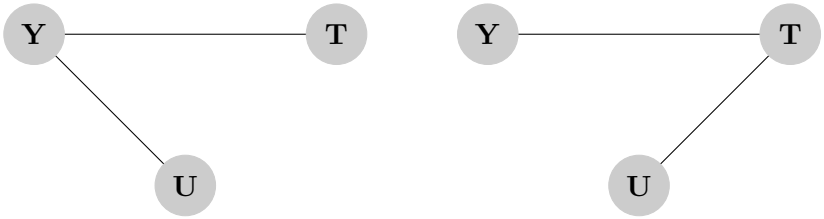


Figure 1: Graphical representation of random selection model (left) and random pattern mixture model (right)

In this circumstance the joint distribution of repeated measurement $\boldsymbol{Y}$, event times $\boldsymbol{T}$ and random effects $\boldsymbol{U}$ can be expressed as

$$\begin{array}{cc} \textit{Random Selection model} & \textit{Random Pattern mixture model} \\ \begin{aligned}[\boldsymbol{Y}, \boldsymbol{T}, \boldsymbol{U}] &= [\boldsymbol{U}][\boldsymbol{Y}|\boldsymbol{U}][\boldsymbol{T}|\boldsymbol{Y}, \boldsymbol{U}] \\ &= [\boldsymbol{U}][\boldsymbol{Y}|\boldsymbol{U}][\boldsymbol{T}|\boldsymbol{Y}]\end{aligned} & \begin{aligned}[\boldsymbol{Y}, \boldsymbol{T}, \boldsymbol{U}] &= [\boldsymbol{U}][\boldsymbol{T}|\boldsymbol{U}][\boldsymbol{Y}|\boldsymbol{T}, \boldsymbol{U}] \\ &= [\boldsymbol{U}][\boldsymbol{T}|\boldsymbol{U}][\boldsymbol{Y}|\boldsymbol{T}]\end{aligned} \end{array}$$

The joint distribution of $\boldsymbol{Y}$, $\boldsymbol{T}$ and $\boldsymbol{U}$ can be visualized in figure 1. The absence of edge indicates conditional independence between two vertices of the edge, given the third vertice involved in the graph. Throughout this dissertation whenever joint model will be considered, we will use selection model.

The primary objective in joint modeling is to find reliable estimates of the parameters involved in the model along with their standard errors. Method of maximum likelihood (ML) estimators is the most widely used estimation procedure. Because of the complicated nature of the likelihood, no closed form solution of the likelihood equation is available. Some iterative procedure (e.g. Newton-Raphson or Fisher's Scoring) is applied to get the estimates of the parameters.

In early literature on joint models, time-to-event data are modeled parametrically, which facilitates, in principle straightforward likelihood inference (Schluchter (1992), Pawitan and Self(1993)). Later on, proportional hazard models are used for time-to-event data. Considerable literature (DeGruttola and Tu (1994), Wulfson and Tsiatis (1197), Henderson *et al. (2000)*, Guo and Carlin (2004), Schluchter (1192), Hogan and Laired (1997) and Ten Have *et al.* (2000)) is available on selection models where the joint density of repeated measures vector and failure time is obtained as the product of the conditional density of the failure time given the longitudinal outcomes and the marginal density of these outcomes. An alternative class of models is called pattern mixture models (Wu and Bailey (1989), Little (1954)) where the longitudinal response is modeled conditional on the survival time and the primary interest is on estimating the parameters of that model. More specifically the selection model would answer the question regarding how one's response on the severity of the disease affects death (failure) whereas the pattern mixture model would demonstrate the pattern of severity of the disease given one's death time.

Joint modeling of survival and longitudinal data has been considered earlier by Wulfson and Tsiatis (1997), Ratcliffe *et al.* (2004), Wu and Carroll (1988) and others. Several attempts have been made to develop a link link between the longitudinal outcomes and the survival data (Henderson *et al.* (2000), Huang *et al.* (2001), Wang and Taylor (2001), Xu and Zeger (2001)). An excellent summary on the use of linear and generalized linear models in analyzing incomplete longitudinal and time-to-event data can be found in Hogan and Laired (1997). More recently, Chakraborty and Das (2010), Chakraborty (2014) have considered MCEM based inference for joint models. Most of the models in these connections assume that subject-level effects are the adequate surrogate for the unobserved factors linking the longitudinal outcomes with the time-to-failure data so that given the subject-level link, the repeated measures and failure times are conditionally independent. This dissertation is organized as follows: Section 2 deals with the model and corresponding methodology. After considering AFT model for time-to-event data and linear mixed effects model for longitudinal data, an MCEM based inference method is proposed. In the next section, the performance of the proposed algorithm is judged by a small sample simulation study. Muscular syndrome data is analyzed in Section 4. This dissertation ends with some discussions.

# 2 The model and methodology

## 2.1 Model for the longitudinal data

It is often modelled by linear mixed-effects model as,

$$y_{ij} = \mu + \alpha x_{ij} + b_i + \epsilon_{ij}; \quad 1 \leq j \leq n_i, \quad 1 \leq i \leq N \tag{1}$$

where $N$ stands for the number of subject involved in a study, $n_i$ stands for the number of longitudinal observations taken on the $ith$ subject, $y_{ij}$ is the response for subject $i$ at time point $j$, $b_i$ is the subject specific random effect, $x_{ij}$ is the realization of a single covariate at time point $j$ and $\epsilon_{ij}$ is the standardized random error.

We have assumed that $b_i \sim N(0, \sigma_b^2)$, $\epsilon_{ij} \sim N(0, 1)$ and they are independently distributed for $j = 1(1)n_i$ moreover $\epsilon_{ij}$s' are i.i.d when $b_i$ is specified.

However, this linear mixed-effects model can accommodate more than one covariates. Sometimes some non-linear model may also be used.

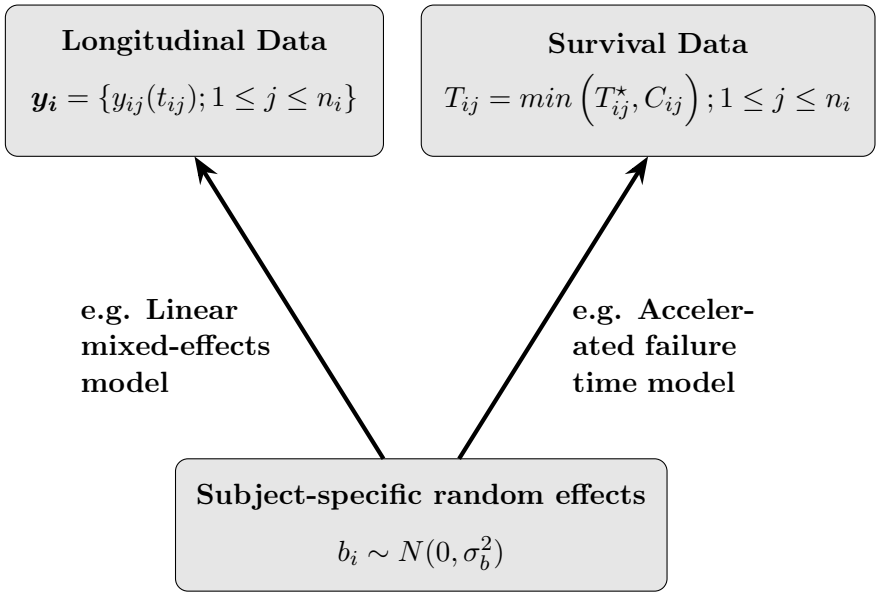


Figure 2: Schematic diagram for joint model

### 2.2 Model for time-to-event

To model the time-to-event data which is the second component of the joint model, Cox Proportional Hazard Model is often used. However, in some situations, as an alternative Accelerated Failure Time (AFT) model can also be used. Beside giving some computational advantage this AFT model have some other advantages too. The choice of AFT model is obviously debatable, however, for the present work we will confine ourselves to AFT model only. For the present work AFT model, with underlying distribution as log-normal, can be expressed as,

$$\log T_{ij} = \beta + b_i \tag{2}$$

i.e. $\log T_{ij} \sim N((\beta + b_i), \sigma_T^2)$, say. $T_{ij}$ is the minimum of actual event time $T_{ij}^{\star}$ and censoring time $C_{ij}$ at time point $j$ for patient $i$. Here we define the censoring indicator variables $\delta_{ij}$ as $\delta_{ij} = I(T_{ij}^{\star} \leq C_{ij})$, i.e.

$$\begin{aligned} \delta_{ij} &= 1; \quad \text{if actual event time is recorded} \\ &= 0; \quad \text{if event time is censored to the right} \end{aligned}$$

We assume that the censoring of survival data and the assessment process of longitudinal measurements are non informative. Here the parameter vector is $\boldsymbol{\theta} = (\mu, \alpha, \beta, \sigma_b)'$.

It is to be noted that both the models (1) and (2) involve the same random effect $b_i$. This subject specific random effect $b_i$ correlates both the models and it is easy to note that given this specific random effect $b_i$, model (1) and model (2) are independent.

### 2.3 Construction of the joint Likelihood function

In this project work we are dealing with selection model i.e. the joint density of observed event time $t_{ij}$ and longitudinal score $y_{ij}$ at time point $j$ is obtained as the product of the conditional density of the event time (or failure time) given the longitudinal outcome and the marginal density of the longitudinal outcome. Hence we obtain the joint likelihood for subject $i$ as follows:

$$\begin{aligned} L_i(\boldsymbol{\theta}) &= \prod_{j=1}^{n_i} f(y_{ij}, \log t_{ij} | \boldsymbol{\theta}) \\ &= \int_{b_i} \prod_{j=1}^{n_i} g_1(\log t_{ij} | y_{ij}, b_i) \times g_2(y_{ij} | b_i) \times \pi(b_i) db_i \\ &= \int_{b_i} \prod_{j=1}^{n_i} g_1(\log t_{ij} | b_i) \times g_2(y_{ij} | b_i) \times \pi(b_i) db_i \\ &= \int_{b_i} l_{iJ} db_i \end{aligned} \tag{3}$$

where $l_{iJ} = \prod_{j=1}^{n_i} g_1(\log t_{ij} | b_i) \times g_2(y_{ij} | b_i) \times \pi(b_i)$.

Finally, we get the complete likelihood as,

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{N} L_i(\boldsymbol{\theta}) \tag{4}$$

### 2.4 Monte Carlo EM (MCEM) based inference

Here the objective is to estimate the model parameters. We use Monte Carlo EM (MCEM) technique to find the maximum likelihood estimates of the parameters.

Let $L(\boldsymbol{\theta})$ be the likelihood of the observed data for all the individuals and $l(\boldsymbol{\theta})$ be the corresponding log-likelihood function. We need to find the score vector and Fisher's information matrix for estimating the parameters involved in the joint model iteratively. Let $\boldsymbol{S_\theta}$ be the overall score vector. Then,

$$\begin{aligned}
\boldsymbol{S_\theta} &= \frac{\partial}{\partial \boldsymbol{\theta}} l(\boldsymbol{\theta}) \\
&= \frac{\partial}{\partial \boldsymbol{\theta}} \log L(\boldsymbol{\theta}) \\
&= \frac{\partial}{\partial \boldsymbol{\theta}} \log \prod_{i=1}^{N} \left\{ \int_{b_i} l_{iJ} db_i \right\} \\
&= \frac{\partial}{\partial \boldsymbol{\theta}} \sum_{i=1}^{N} \log \left\{ \int_{b_i} l_{iJ} db_i \right\} \\
&= \sum_{i=1}^{N} \frac{1}{\left\{ \int_{b_i} l_{iJ} db_i \right\}} \times \frac{\partial}{\partial \boldsymbol{\theta}} \left\{ \int_{b_i} l_{iJ} db_i \right\} \\
&= \sum_{i=1}^{N} \frac{1}{\left\{ \int_{b_i} l_{iJ} db_i \right\}} \times \left\{ \int_{b_i} \frac{\partial}{\partial \boldsymbol{\theta}} l_{iJ} db_i \right\} \\
&= \sum_{i=1}^{N} \frac{1}{\left\{ \int_{b_i} l_{iJ} db_i \right\}} \times \left\{ \int_{b_i} \frac{\partial}{\partial \boldsymbol{\theta}} \log l_{iJ} \times l_{iJ} db_i \right\} \\
&= \sum_{i=1}^{N} \left\{ \int_{b_i} \left( \frac{\partial}{\partial \boldsymbol{\theta}} \log l_{iJ} \right) \times \frac{l_{iJ}}{\int_{b_i} l_{iJ} db_i} db_i \right\} \qquad (5)
\end{aligned}$$

Now,

$$\begin{aligned}
\frac{l_{iJ}}{\int_{b_i} l_{iJ} db_i} &= \frac{\prod_{j=1}^{n_i} g_1(\log t_{ij}|b_i) \times g_2(y_{ij}|b_i) \times \pi(b_i)}{\int_{b_i} \prod_{j=1}^{n_i} g_1(\log t_{ij}|b_i) \times g_2(y_{ij}|b_i) \times \pi(b_i) db_i} \\
\Rightarrow h\left(b_i|y_{ij}, \log t_{ij}\right) &\propto \prod_{j=1}^{n_i} g_1(\log t_{ij}|b_i) \times g_2(y_{ij}|b_i) \times \pi(b_i) \qquad (6) \\
&= \text{conditional density of } b_i \text{ given data}
\end{aligned}$$

Hence, from (5),

$$\begin{aligned}
\boldsymbol{S_\theta} &= \sum_{i=1}^{N} \left\{ \int_{b_i} \left( \frac{\partial}{\partial \boldsymbol{\theta}} \log l_{iJ} \right) \times h\left(b_i | y_{ij}, \log t_{ij}\right) db_i \right\} \\
&= \sum_{i=1}^{N} E_{b_i | y_{ij}, \log t_{ij}} \left( \frac{\partial}{\partial \boldsymbol{\theta}} \log l_{iJ} \right) \\
&= \sum_{i=1}^{N} E_{b_i | \text{data}} \left( \frac{\partial}{\partial \boldsymbol{\theta}} \log l_{iJ} \right) \\
&= \sum_{i=1}^{N} E_{b_i | \text{data}} \left( \boldsymbol{S^\star_{\theta i}} \right); \quad \boldsymbol{S^\star_{\theta i}} \text{ is conditional score vector for subject } i \quad (7)
\end{aligned}$$

In the proposed algorithm, given $y_{ij}$ and $\log t_{ij}$, unknown $b_i$'s are simulated from equation (6) using the Metropolis Sampler taking some suitable proposal. These $b_i$'s are then used to compute $E_{b_i|\text{data}}\left(\boldsymbol{S^\star_{\theta i}}\right)$ which is approximated by Monte Carlo method i.e.

$$E_{b_i | y_{ij}, \log t_{ij}} \left( \boldsymbol{S^\star_{\theta i}} \right) \simeq \frac{1}{(R - burnin)} \sum_{\ell=1}^{(R-burnin)} \boldsymbol{S^\star_{\theta i}}^{(\ell)} \quad (8)$$

where $R$ (here 10000), is the number of random observations generated initially by the Metropolis sampler and *burnin* is the size of the burnt in sample, taken as 10 per cent of $R$. Therefore using (7) we have the approximate expression for the overall score vector $\boldsymbol{S_\theta}$ as,

$$\boldsymbol{S_\theta} \simeq \sum_{i=1}^{N} \left( \frac{1}{(R - burnin)} \sum_{\ell=1}^{(R-burnin)} \boldsymbol{S^\star_{\theta i}}^{(\ell)} \right) \quad (9)$$

The overall information matrix is obtained by differentiating $\boldsymbol{S_\theta}$ with respect to $\boldsymbol{\theta}$, multiplying it with a minus one and finally taking its expectation i.e.

$$\boldsymbol{I(\theta)} = E_{\boldsymbol{\theta}} \left( -\frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{S_\theta} \right)$$

and in a similar fashion, as in (6), it can also be expressed as the sum of conditional information matrices as follows,

$$
\begin{aligned}
-\frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{S_\theta} &= -\sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} E_{b_i|y_{ij},\log t_{ij}} \left( \boldsymbol{S^\star_{\theta i}} \right) \\
\Rightarrow \boldsymbol{I(\theta)} &= \sum_{i=1}^{N} E_{b_i|y_{ij},\log t_{ij}} \left( -\frac{\partial}{\partial \boldsymbol{\theta}} \boldsymbol{S^\star_{\theta i}} \right) \\
\Rightarrow \boldsymbol{I(\theta)} &= \sum_{i=1}^{N} \boldsymbol{I(\theta)^\star_i} \qquad (10)
\end{aligned}
$$

From (6), expression for the conditional density of $h\left(b_i|y_{ij}, \log t_{ij}\right)$ without the normality constant can be given by the help of individual densities as follows,

$$
\begin{aligned}
g_1(\log t_{ij}|b_i) &\propto \frac{1}{\sigma_T} exp\left\{ -\frac{1}{2\sigma_T^2} (\log t_{ij} - \beta - b_i)^2 \right\} \\
g_2(y_{ij}|b_i) &\propto exp\left\{ -\frac{1}{2} (y_{ij} - \mu - \alpha x_{ij} - b_i)^2 \right\} \\
\pi(b_i) &\propto \frac{1}{\sigma_b} exp\left\{ -\frac{1}{2\sigma_b^2} b_i^2 \right\} \qquad (11)
\end{aligned}
$$

So that,

$$
\begin{aligned}
h\left(b_i|y_{ij}, \log t_{ij}\right) \propto \prod_{j=1}^{n_i} & \left\{ \frac{1}{\sigma_T} \phi\left( \frac{\log t_{ij} - \beta - b_i}{\sigma_T} \right) \right\}^{\delta_{ij}} \\
& \times \left\{ 1 - \Phi\left( \frac{\log c_{ij} - \beta - b_i}{\sigma_T} \right) \right\}^{1-\delta_{ij}} \\
& \times \phi\left( y_{ij} - \mu - \alpha x_{ij} - b_i \right) \\
& \times \frac{1}{\sigma_b} \phi\left( \frac{b_i}{\sigma_b} \right) \qquad (12)
\end{aligned}
$$

The main challenge in the likelihood inference for joint models is the computational complexity, since numerical methods or Monte Carlo methods can be very computationally intensive when the dimension of the random effects is not small. Moreover, convergence of the EM algorithms can sometimes be an issue. Computations of score function and information matrix are given in Appendix A.

### 2.5 MCEM algorithm in a nutshell

**Step 1:** Choose an initial estimate of $\theta$, say, $\theta^{(0)}$ (using some suitable techniques)

**Step 2:** Set $k = 0$

**Step 3:** Generate $R$ (10000, say) observations from $h(b_i|y_{ij}, \log t_{ij})$ using Metropolis sampler (**a lot of difficulties involved!**) taking some suitable proposal (Markov Chain)

**Step 4:** Approximate the expectations involved in score function and information matrix (E-step with Monte Carlo)

**Step 5:** Update the estimates using scoring method as,

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \boldsymbol{I}\left(\boldsymbol{\theta}^{(\boldsymbol{k})}\right)^{-\mathbf{1}} \boldsymbol{S}\left(\boldsymbol{\theta}^{(\boldsymbol{k})}\right)$$

(M-step with Fisher's Scoring)

**Step 6:** Set $k = k + 1$

**Step 7:** Continue **Step 2-6** till convergence

## 3 Simulation studies

In this section we have performed several sets of simulation studies with different choices of true values of the parameters. We have fixed the number of subjects $N$ at 25 and there are $n = 4$ longitudinal observations on each of them. Three different choices of $\sigma_T$ are used for the simulation studies. To find initial estimates, we have fitted both the sub models separately. For linear mixed-effects model, we have used the R package `nlme` (http://cran.r-project.org/web/packages/nlme/index.html) and for accelerated failure time model we have used the package `survival` (http://cran.r-project.org/web/packages/survival/index.html). These estimates are taken to be the initial estimates which make the convergence faster. Corresponding R codes can be found in Appendix B.

**Simulation: 1**

Data generated using $\sigma_T = 1.55$.

Table 1: Parameter estimates obtained using MCEM algorithm

| Parameter | True value | Estimate | Standard Error | Absolute bias |
|---|---|---|---|---|
| $\mu$ | -3 | -4.504973 | 0.53672034 | 1.504973 |
| $\alpha$ | 2 | 2.010104 | 0.01294196 | 0.010104 |
| $\beta$ | 2.5 | 1.297291 | 0.59741468 | 1.202709 |
| $\sigma_b$ | 1.225 | 1.467351 | 0.25016115 | 0.242606 |

Convergence patterns for various parameters obtained using MCEM algorithm is presented in figure 3.

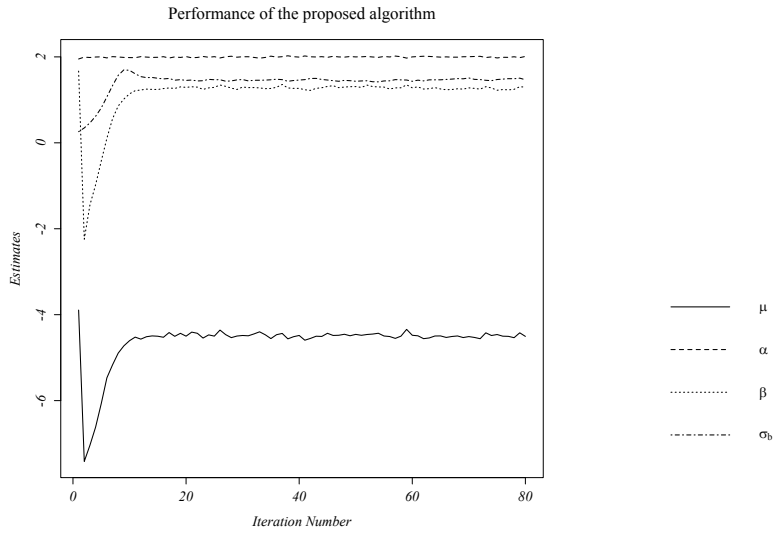


Figure 3: Convergence patterns of various parameters

From figure 3, it can be seen that the estimate of $\alpha$ shows very constant pattern, where as estimates for $\mu$ and $\beta$ show a lot of hiccup at early stages. However for all parameters the method converges though $\mu$ and $\beta$ show relatively large bias compared to $\alpha$ and $\sigma_b$. The standard errors are really within the acceptance level.

**Simulation: 2**
Data generated using $\sigma_T = 0.50$.

Table 2: Parameter estimates obtained using MCEM algorithm

| Parameter | True value | Estimate | Standard Error | Absolute bias |
|---|---|---|---|---|
| $\mu$ | -3 | -4.635835 | 0.14281614 | 1.6358 |
| $\alpha$ | 2 | 2.061284 | 0.01618356 | 0.061284 |
| $\beta$ | 2.5 | 1.294264 | 0.12623253 | 1.2057 |
| $\sigma_b$ | 1.225 | 1.346767 | 0.04300177 | 0.1218 |

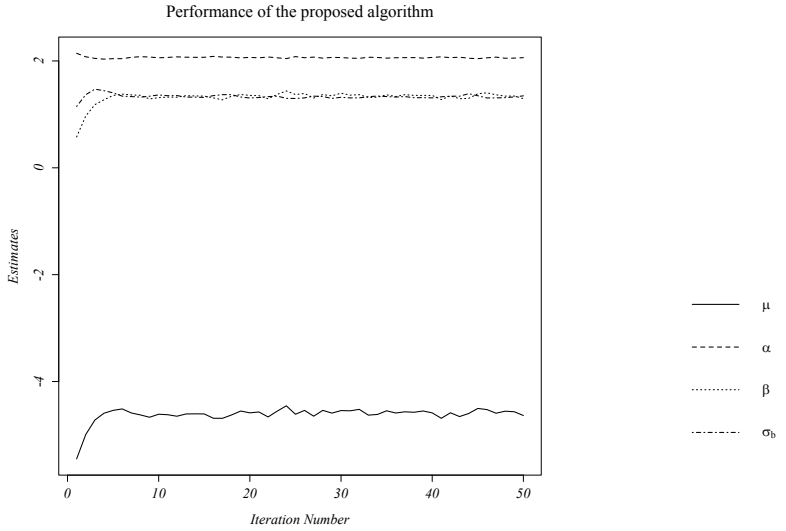


Figure 4: Convergence patterns of various parameters

**Simulation 3:**
Data generated using $\sigma_T = 0.15$.

Table 3: Parameter estimates obtained using MCEM algorithm

| Parameter | True value | Estimate | Standard Error | Absolute bias |
|---|---|---|---|---|
| $\mu$ | -3 | -4.229542 | 0.07972108 | 1.2295 |
| $\alpha$ | 2 | 2.132087 | 0.01024069 | 0.132087 |
| $\beta$ | 2.5 | 2.002448 | 0.07776745 | 0.4976 |
| $\sigma_b$ | 1.225 | 1.507479 | 0.07079735 | 0.2857 |

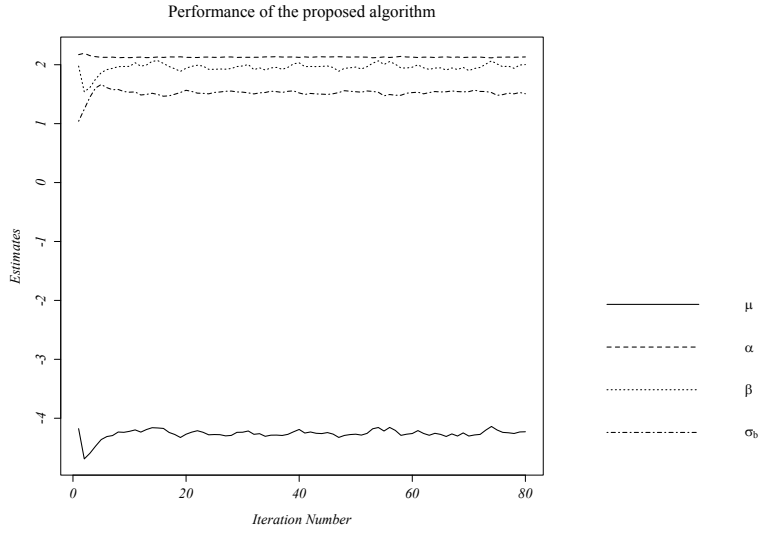


Figure 5: Convergence patterns of various parameters

It is to be noted that for third simulation i.e. when we considered $\sigma_T = 0.15$, the proposed algorithm performed very well. In this case only the parameter $\mu$ has absolute bias bigger than 1 while rest of the parameters have absolute biases very small.

This phenomena indicates that proposed algorithm gives good estimates under small variation of the time-to-event process.

# 4 Application to Muscular Dystrophy Syndrome data

This work is motivated by a data set obtained from National Neurosciences Centre, Kolkata (http://www.nnccalcutta.com/). The data consists of muscle scores of children suffering from a muscular dystrophy syndrome. This is a genetic disease affecting only male children. Over different time points, composite muscle scores (average of ten muscle scores) are considered to be an indicator of quality of life of these patients. A score near 10 is assumed to be a good one. However, among 26 patients, only 19 patients are considered for this analysis with at least 4 observations. Along with the composite muscle score, time taken by a patient to walk 4 steps is also noted. For this time-to-event data, time beyond 60 seconds is assumed to be censored. A patient with high muscle score can easily walk 4 steps. This motivates us to use a selection model since the time-to-event data depends on the longitudinal outcome.

Table 4: Partial Data only for two subjects

| | Patient | Score | Climbing_Stair | Age_at_time | Onset_age |
|---|---|---|---|---|---|
| 1 | P1 | 4.33 | 6 | 7 | 4 |
| 2 | P1 | 4.83 | 5 | 7 | 4 |
| 3 | P1 | 3.50 | 900 | 10 | 4 |
| 4 | P1 | 3.00 | 900 | 11 | 4 |
| 5 | P2 | 3.17 | 900 | 8 | 3 |
| 6 | P2 | 3.50 | 900 | 8 | 3 |
| 7 | P2 | 3.17 | 900 | 8 | 3 |
| 8 | P2 | 2.83 | 900 | 8 | 3 |

A data frame with 76 rows and 5 columns. Each of 4 successive rows represents data for the same subject or patient.

- `Patient`: A factor with levels $P1, P2, \ldots, P19$. Each level represents a particular patient. There are 19 patients in aggregate.

- `Score`: Composite muscle score (average of ten muscle scores).

- `Climbing_Stair`: Time (in seconds) taken by a patient walk 4 steps. Time beyond 60 seconds is assumed to be censored. A value 900 indicates time is censored to the right.

- `Age_at_time`: Age of the patient at the time of study.

- `Onset_age`: Age of a patient when the disease on him/her was first discovered. This is treated as the only covariate.

For the composite score, a linear mixed effects model is used as,

$$\texttt{Score}_{ij} = \mu + \alpha * \texttt{Onset\_age}_{ij} + b_i + \epsilon_{ij}$$

and for time-to-event, an AFT model is used as,

$$\log(\texttt{Climbing\_Stair}_{ij}) = \beta + b_i$$

The objective is to obtain the estimates of the parameters $\mu$, $\alpha$, $\beta$ and $\sigma_b$ by applying the joint model described in section 4. Summary of the analysis is given below:

Table 5: Parameter estimates applying the Joint Model

| Parameter | Estimate | Standard Error |
|---|---|---|
| $\mu$ | 3.61647686 | 0.20449250 |
| $\alpha$ | 0.23480275 | 0.04372608 |
| $\beta$ | 3.96921363 | 0.07389875 |
| $\sigma_b$ | 0.01304504 | 0.28348920 |

Convergence patterns for various parameters obtained using MCEM algorithm is presented in figure 6.

From figure 6 it can be seen that initially there are a few hiccups for almost all parameters. They eventually converge as the number of iteration increased. The proposed MCEM performed incredibly well.

## 5 Discussion

Principal objective of this work is to obtain the estimates of the parameters for a data set on muscular dystrophy syndrome. In simulation studies, we have noticed that MCEM performed well for fixed values of $\sigma_T$ i.e. the standard deviation of time-to-event variable. But when considering $\sigma_T$ as random, sometimes the algorithm produces negative estimate of $\sigma_T$ . It is not very strange as it can be judged theoretically in case of variance estimation problem. It needs further investigation. To judge the performance of the proposed algorithm coverage probabilities may also be calculated.

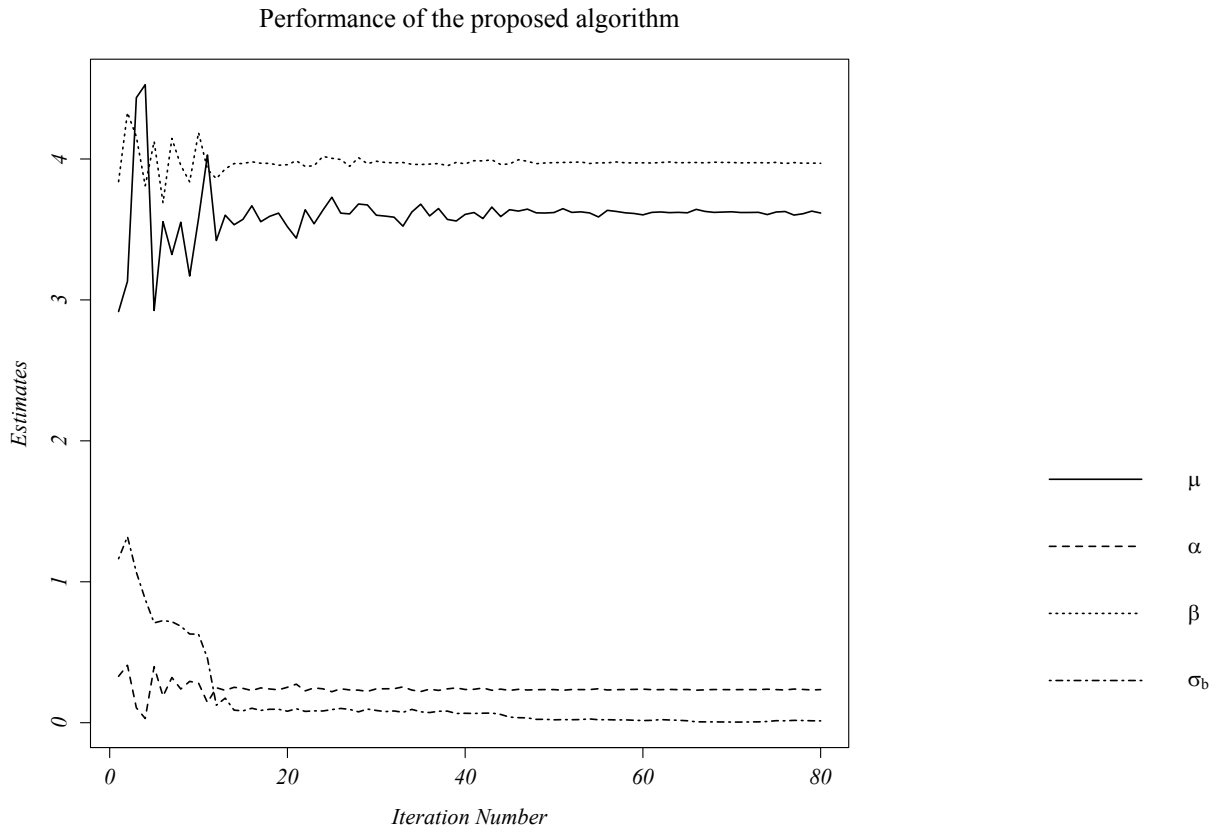


Figure 6: Convergence patterns of various parameters

# A Derivation of score function and information matrix of the joint likelihood

$$\frac{\partial}{\partial \mu} \log l_{iJ} = \sum_{j=1}^{n_i} (y_{ij} - \mu - \alpha x_{ij} - b_i) \tag{13}$$

$$\frac{\partial}{\partial \alpha} \log l_{iJ} = \sum_{j=1}^{n_i} x_{ij}(y_{ij} - \mu - \alpha x_{ij} - b_i) \tag{14}$$

$$\frac{\partial}{\partial \beta} \log l_{iJ} = \sum_{j=1}^{n_i} \frac{\delta_{ij}}{\sigma_T} \left( \frac{\log t_{ij} - \beta - b_i}{\sigma_T} \right) + \frac{1 - \delta_{ij}}{\sigma_T} \left\{ \frac{\phi\left( \frac{\log c_{ij} - \beta - b_i}{\sigma_T} \right)}{1 - \Phi\left( \frac{\log c_{ij} - \beta - b_i}{\sigma_T} \right)} \right\} \tag{15}$$

$$\frac{\partial}{\partial \sigma_b} \log l_{iJ} = \sum_{j=1}^{n_i} \left( -\frac{1}{\sigma_b} + \frac{b_i^2}{\sigma_b^3} \right) \tag{16}$$

$$\frac{\partial^2}{\partial \mu^2} \log l_{iJ} = -n_i \tag{17}$$

$$\frac{\partial^2}{\partial \mu \partial \alpha} \log l_{iJ} = -\sum_{j=1}^{n_1} x_{ij} \tag{18}$$

$$\frac{\partial^2}{\partial \mu \partial \beta} \log l_{iJ} = 0 \tag{19}$$

$$\frac{\partial^2}{\partial \mu \partial \sigma_b} \log l_{iJ} = 0 \tag{20}$$

$$\frac{\partial^2}{\partial \alpha^2} \log l_{iJ} = -\sum_{j=1}^{n_i} x_{ij}^2 \tag{21}$$

$$\frac{\partial^2}{\partial \alpha \partial \beta} \log l_{iJ} = 0 \tag{22}$$

$$\frac{\partial^2}{\partial \alpha \partial \sigma_b} \log l_{iJ} = 0 \tag{23}$$

$$\frac{\partial^2}{\partial\beta^2}\log l_{iJ} = \sum_{j=1}^{n_i} -\frac{\delta_{ij}}{\sigma_T^2} + \frac{1-\delta_{ij}}{\sigma_T^2}\left\{\frac{\phi\left(\frac{\log c_{ij}-\beta-b_i}{\sigma_T}\right)}{1-\Phi\left(\frac{\log c_{ij}-\beta-b_i}{\sigma_T}\right)}\right\} \times\left[\left(\frac{\log c_{ij}-\beta-b_i}{\sigma_T}\right)-\left\{\frac{\phi\left(\frac{\log c_{ij}-\beta-b_i}{\sigma_T}\right)}{1-\Phi\left(\frac{\log c_{ij}-\beta-b_i}{\sigma_T}\right)}\right\}\right] \tag{24}$$

$$\frac{\partial^2}{\partial\beta\partial\sigma_b}\log l_{iJ} = 0 \tag{25}$$

$$\frac{\partial^2}{\partial\sigma_b^2}\log l_{iJ} = \sum_{j=1}^{n_i}\left(\frac{1}{\sigma_b^2}-\frac{3b_i^2}{\sigma_b^4}\right) \tag{26}$$

Now, the conditional score vector is given by,

$$\boldsymbol{S}^{\star}_{\boldsymbol{\theta i}} = \begin{pmatrix} \frac{\partial}{\partial\mu}\log l_{iJ} \\ \frac{\partial}{\partial\alpha}\log l_{iJ} \\ \frac{\partial}{\partial\beta}\log l_{iJ} \\ \frac{\partial}{\partial\sigma_b}\log l_{iJ} \end{pmatrix} \tag{27}$$

and also the conditional information matrix by,

$$\boldsymbol{I}(\boldsymbol{\theta})^{\star}_{\boldsymbol{i}} = \begin{pmatrix} \frac{\partial^2}{\partial\mu^2}\log l_{iJ} & \frac{\partial^2}{\partial\mu\partial\alpha}\log l_{iJ} & \frac{\partial^2}{\partial\mu\partial\beta}\log l_{iJ} & \frac{\partial^2}{\partial\mu\partial\sigma_b}\log l_{iJ} \\ & \frac{\partial^2}{\partial\alpha^2}\log l_{iJ} & \frac{\partial^2}{\partial\alpha\partial\beta}\log l_{iJ} & \frac{\partial^2}{\partial\alpha\partial\sigma_b}\log l_{iJ} \\ & & \frac{\partial^2}{\partial\beta^2}\log l_{iJ} & \frac{\partial^2}{\partial\beta\partial\sigma_b}\log l_{iJ} \\ & & & \frac{\partial^2}{\partial\sigma_b^2}\log l_{iJ} \end{pmatrix} \tag{28}$$

## B R code for simulation studies

Simulation study is done with the help of R package (*ver. 3.2.0, "Full of Ingredients"*), which can be found at http://cran.r-project.org/. R Code for reproducing the simulation and graphics is presented below.

```
## Joint Model for Longitudinal & time-to-event data

rm(list = ls())
library(survival)
N <- 25; n <- 4;

##-------TRUE VALUES for generating hypothetical data------##
mu.star <- -3
alpha.star <- 2
b.eta.star <- 2.5
sig.b.star <- sqrt(1.5)
sig.t <- 1.55 #constant
##-----------------------------------------------------------##

## covariate matrix
x <- matrix(rnorm(n * N, 3, sqrt(2)), N, n, byrow = TRUE)
rownames(x) <- paste("P:",1:25)
colnames(x) <- c("Jan", "Feb", "Mar", "Apr")

## Matrix of all 1's
J <- matrix(1, N, n, byrow = TRUE)

## Matrix of random effects with
## identical columns (unconditional pdf of b)
B.RAND <- matrix(rnorm(N, 0, sig.b.star), N, n)

##---------------Longitudinal data generation-------------##
y <- mu.star * J +
  alpha.star * x +
  B.RAND +
  matrix(rnorm(N*n, 0, 1), N, n, byrow = T)

##---------------Survival data generation----------------##
tau <- 40
truetime <- exp(b.eta.star + B.RAND) #actual time
#censored time
cstime <- matrix(runif(N*n, 0, tau), N, n, byrow = T)
```

```
time <- pmin(truetime, cstime) #observed time
event <- time == truetime
del <- event * 1
## log time points beyond which censoring takes place
ln.C <- log(cstime)
## Vector of log(observed event-times)
ln.T <- log(time)
(p <- length(event[event == F]) / (N*n))

## Initial choices for MCEM Algorithm
Y <- as.vector(t(y))
X <- as.vector(t(x))
fit.1 <- lm(Y ~ X)
mu <- fit.1[["coefficients"]][["(Intercept)"]]
alpha <- fit.1[["coefficients"]][["X"]]
##-------------------------------------------------------##
Time <- as.vector(t(time)) #follow-up time vector
Del <- as.vector(t(del))
survobj <- Surv(Time, Del) #survival object
#AFT model with no covariate
aft.fit <- survreg(survobj ~ 1,dist = "lognormal")
b.eta <- aft.fit[["coefficients"]][["(Intercept)"]]
##-------------------------------------------------------##
sig.b <- sd(B.RAND[,1])
sig.t <- 1.55 #constant

(theta.ini <- c(mu, alpha, b.eta, sig.b))
##-------------------------------------------------------##
Q <- function(theta.ini)
{
  mu <- theta.ini[1]
  alpha <- theta.ini[2]
  b.eta <- theta.ini[3]
  sig.b <- theta.ini[4]

  ## Functions of differentials ##
  ##-----------------------------------------------------##
  e1 <- function(x, y, b){y - mu - alpha * x - b}
  e2 <- function(ln.t,b){(ln.t - b.eta - b)/sig.t}
  e3 <- function(ln.c,b){(ln.c - b.eta - b)/sig.t}

  d.mu.1 <- function(x, y, b)
```

```
{
  sum(e1(x, y, b))
}

d.alpha.1 <- function(x, y, b)
{
  sum(x * e1(x, y, b))
}

d.b.eta.1 <- function(ln.t, ln.c, b, delta)
{
  sum((delta * sig.t**(-1) * e2(ln.t,b)) +
        (1 - delta) * sig.t**(-1) *
        dnorm(e3(ln.c,b)) *
        pnorm(e3(ln.c,b),lower.tail = F)**(-1))
}

d.sig.b.1 <- function(b)
{
  sum(-sig.b**(-1) + (b**2 * sig.b**(-3)))
}

##-----------------------------------------------------##
d.mu.2 <- -n

d.mualpha.2 <- function(x)
{
  -sum(x)
}

d.mub.eta.2 <- 0

d.musig.b.2 <- 0

d.alpha.2 <- function(x)
{
  -sum(x**2)
}

d.alphab.eta.2 <- 0

d.alphasig.b.2 <- 0
```

```
d.b.eta.2 <- function(ln.c, b, delta)
{
  sum(-delta * sig.t**(-2) +
        (1-delta) * sig.t**(-2) * e3(ln.c,b) *
        dnorm(e3(ln.c,b)) *
        pnorm(e3(ln.c,b),lower.tail = F)**(-1) -
        (1-delta) * sig.t**(-2) * dnorm(e3(ln.c,b))**2 *
        pnorm(e3(ln.c,b),lower.tail = F)**(-2))
}

d.b.etasig.b.2 <- 0

d.sig.b.2 <-function(b)
{
  sum(sig.b**(-2) - 3 * b**2 * sig.b**(-4))
}

## Metropolis sampler for simulating b's for each i
rrandeff <- function(Nsim)
{
  h <- function(b)
  {
    prod(dnorm(b/sig.b) *
           dnorm((ln.T[i,] - b.eta - b)/sig.t)**del[i,] *
           pnorm(((ln.C[i,] - b.eta - b)/sig.t),
           lower.tail = F)**
           (1-del[i,]) *
           dnorm((y[i,] - mu - alpha*x[i,] - b)))
  }

  t <- 1
  s <- rep(0, Nsim)
  s[1] <- rnorm(1, 5, sig.b) #initialization

  for(t in 2:Nsim)
  {
    sprime <- rnorm(1, 5, sig.b) #candidate observation
    a <- h(sprime)/h(s[t-1])
    if(is.nan(a)){a <- 0; nan <- 1}
    acprob <- min(1, a) #acceptance probability
    u <- runif(1, 0, 1)
```

```
    if(u < acprob) s[t] <- sprime else s[t] <- s[t-1]
  }
  return(s)
}

##-------------------------------------------------------##

#Unconditional Score Vector
UCOND.S <- rep(0, 4)
#Unconditional Information Matrix
UCOND.I <- array(0, dim = c(4, 4))

R <- 10000
burn <- 1000

for(i in 1:N)
{
  COND.b <- rrandeff(R)
  ##-------------------------------------------------------##
  COND.S <- rep(0, 4)
  COND.I <- array(0, dim = c(4, 4))
  D <- array(0, dim = c(14, (R - burn)))
  for(j in 1:ncol(D))
  {
    D[,j] <- c(d.mu.1(x[i,], y[i,], COND.b[burn + j]),
               d.alpha.1(x[i,], y[i,],
                         COND.b[burn + j]),
               d.b.eta.1(ln.T[i,], ln.C[i,],
                        COND.b[burn + j], del[i,]),
               d.sig.b.1(COND.b[burn + j]),
               ##-------------------------------------##
               d.mu.2,
               d.mualpha.2(x[i,]),
               d.mub.eta.2,
               d.musig.b.2,
               d.alpha.2(x[i,]),
               d.alphab.eta.2,
               d.alphasig.b.2,
               d.b.eta.2(ln.C[i,], COND.b[burn + j],
                        del[i,]),
               d.b.etasig.b.2,
               d.sig.b.2(COND.b[burn + j])
```

```
      )
    }
    W <- apply(D, 1, mean)
    COND.S <- W[1:4] #Conditional Score Vector
    UCOND.S <- UCOND.S + COND.S

    ##---------------------------------------------------##
    COND.I[1,1] <- W[5]
    COND.I[1,2] <- COND.I[2,1] <- W[6]
    COND.I[1,3] <- COND.I[3,1] <- W[7]
    COND.I[1,4] <- COND.I[4,1] <- W[8]
    COND.I[2,2] <- W[9]
    COND.I[2,3] <- COND.I[3,2] <- W[10]
    COND.I[2,4] <- COND.I[4,2] <- W[11]
    COND.I[3,3] <- W[12]
    COND.I[3,4] <- COND.I[4,3] <- W[13]
    COND.I[4,4] <- W[14] #Conditional Information Matrix
    ##---------------------------------------------------##
    UCOND.I <- UCOND.I + COND.I
  }
  ucond <- list(score = UCOND.S, inf = -UCOND.I)
  return(ucond)
}

niter <- 80 # no. of iteration
res <- array(0, dim = c(niter,4))
colnames(res) <- c("mu", "alpha", "b.eta", "sig.b")

for(k in 1:niter)
{
  theta <- theta.ini + solve(Q(theta.ini)$inf) %*%
    Q(theta.ini)$score

  ## Updating estimates...
  mu <- theta[1]
  alpha <- theta[2]
  b.eta <- theta[3]
  sig.b <- theta[4]
  theta.ini <- c(mu, alpha, b.eta, sig.b)
  res[k,] <- theta.ini
  print(res[k,])
}
```

```
ul <- max(apply(res, 2, max)) # y-axis upper limit
ll <- min(apply(res, 2, min)) # y-axis lower limit

par(mar=par()$mar+c(0,0,0,3.5), xpd = T, cex = 0.95,
    family = "serif",
    font.axis = 3, font.lab = 3, font.main = 1)

plot(1:niter, res[,1], ylim = c(ll, ul), type = "l",
     lty = 1, lwd = 1.5,
     main = "Performance of the proposed algorithm",
     xlab = "Iteration Number",
     ylab = "Estimates")
lines(1:niter, res[,2], lty = 2, lwd = 1.5)
lines(1:niter, res[,3], lty = 3, lwd = 1.5)
lines(1:niter, res[,4], lty = 4, lwd = 1.5)
legend("bottomright", inset=c(-0.5,0),
       legend = c(expression(mu),
                  expression(alpha),
                  expression(beta),
                  expression(sigma[b])),
       lty = 1:4, lwd = rep(1.5,4), bty = "n")

## Standard error of the estimates
std.err <- apply(res, 2, sd)

##------------------------END-----------------------------##
```

# C R Code for data preparation and analysis

```
setwd("C:\\Users\\Koushik\\Dropbox\\M.Sc_Project\\Project~Code")
rm(list = ls(all = TRUE))
library(nlme)
library(survival)
N <- 19; n <- 4
#********************** Data Preparation ********************
DMD <- read.csv("DMD.csv", header = T)
#View(DMD)
attach(DMD)
p <- unique(Patient)
v <- NULL
##############################################################
for(i in p)
{
  print(i)
  u <- subset(DMD, Patient == i)
  v <- c(v, length(u$Patient)) # vector of no. of patients
}
#v
max.len <- max(v)
##############################################################

## For muscle score
y <- matrix(0, nrow = length(p),
            ncol = max.len, byrow = T)
rownames(y) <- p
for(i in p)
{
  u <- subset(DMD, Patient == i)
  y[i,] <- c(u$Score,
             rep(NA, max.len - length(u$Patient)))
}

## For onset age
x <- matrix(0,
            nrow = length(p), ncol = max.len, byrow = T)
rownames(x) <- p
for(i in p)
{
  u <- subset(DMD, Patient == i)
```

```
  x[i,] <- c(u$Onset_age,
             rep(NA, max.len - length(u$Patient)))
}

## For Climbing stair
time <- matrix(0, nrow = length(p),
               ncol = max.len, byrow = T)
rownames(time) <- p
for(i in p)
{
  u <- subset(DMD, Patient == i)
  time[i,] <- c(u$Climbing_Stair,
                rep(NA, max.len - length(u$Patient)))
}

detach(DMD)

## Censoring indicator
del <- matrix(0, nrow = 19, ncol = 4)
for(i in 1:nrow(time))
{
  for(j in 1:ncol(time))
  {
    if(time[i,j] == 900)
      del[i,j] <- 0
    else del[i,j] <- 1
  }
}

#********************** Data Analysis ***********************
ln.T <- log(time)
ln.C <- log(matrix(60, nrow = N, ncol = n))

p1 <- rep(p, each = 4) # subject levels
data.test <- data.frame(Subject = p1,
                        LongScore = as.vector(t(y)),
                        OnsetAge = as.vector(t(x)))
data.test <- groupedData(LongScore ~ OnsetAge | Subject,
                         data = data.test)
fit.1 <- lme(LongScore ~ OnsetAge, data = data.test,
             random = ~ 1, method = "ML")
```

```
## Initial choices for MCEM Algorithm
mu <- 4.37499738
alpha <- 0.0833338
##-------------------------------------------------------##
#follow-up time vector
Time <- as.vector(t(time))
Del <- as.vector(t(del))
#survival object
survobj <- Surv(Time, Del)

#AFT model with no covariate
aft.fit <- survreg(survobj ~ 1,dist = "lognormal")
b.eta <- aft.fit[["coefficients"]][["(Intercept)"]]
##-------------------------------------------------------##
sig.b <- 1.259261
sig.t <- 4.45 #constant

(theta.ini <- c(mu, alpha, b.eta, sig.b))
##-------------------------------------------------------##
Q <- function(theta.ini)
{
  mu <- theta.ini[1]
  alpha = theta.ini[2]
  b.eta = theta.ini[3]
  sig.b = theta.ini[4]

  ## Functions of differentials ##
  ##-----------------------------------------------------##
  e1 <- function(x, y, b){y - mu - alpha * x - b}
  e2 <- function(ln.t,b){(ln.t - b.eta - b)/sig.t}
  e3 <- function(ln.c,b){(ln.c - b.eta - b)/sig.t}

  d.mu.1 <- function(x, y, b)
  {
    sum(e1(x, y, b))
  }

  d.alpha.1 <- function(x, y, b)
  {
    sum(x * e1(x, y, b))
  }
```

```
d.b.eta.1 <- function(ln.t, ln.c, b, delta)
{
  sum((delta * sig.t**(-1) * e2(ln.t,b)) +
        (1 - delta) * sig.t**(-1) *
        dnorm(e3(ln.c,b)) *
        pnorm(e3(ln.c,b),lower.tail = F)**(-1))
}

d.sig.b.1 <- function(b)
{
  sum(-sig.b**(-1) + (b**2 * sig.b**(-3)))
}

##-----------------------------------------------------##
d.mu.2 <- -n

d.mualpha.2 <- function(x)
{
  -sum(x)
}

d.mub.eta.2 <- 0

d.musig.b.2 <- 0

d.alpha.2 <- function(x)
{
  -sum(x**2)
}

d.alphab.eta.2 <- 0

d.alphasig.b.2 <- 0

d.b.eta.2 <- function(ln.c, b, delta)
{
  sum(-delta * sig.t**(-2) +
        (1-delta) * sig.t**(-2) * e3(ln.c,b) *
        dnorm(e3(ln.c,b)) *
        pnorm(e3(ln.c,b),lower.tail = F)**(-1) -
        (1-delta) * sig.t**(-2) * dnorm(e3(ln.c,b))**2 *
```

```
        pnorm(e3(ln.c,b),lower.tail = F)**(-2))
}

d.b.etasig.b.2 <- 0

d.sig.b.2 <-function(b)
{
  sum(sig.b**(-2) - 3 * b**2 * sig.b**(-4))
}

## Metropolis sampler for simulating b's for each i
rrandeff <- function(Nsim)
{
  h <- function(b)
  {
    prod(dnorm(b/sig.b) *
           dnorm((ln.T[i,] - b.eta - b)/sig.t)**del[i,] *
           pnorm(((ln.C[i,] - b.eta - b)/sig.t),
           lower.tail = F)**
           (1-del[i,]) *
           dnorm((y[i,] - mu - alpha*x[i,] - b)))
  }

  t <- 1
  s <- rep(0, Nsim)
  s[1] <- rnorm(1, 0, sig.b) #initialization

  for(t in 2:Nsim)
  {
    sprime <- rnorm(1, 10, sig.b) #candidate observation
    a <- h(sprime)/h(s[t-1])
    if(is.nan(a)){a <- 0; nan <- 1}
    acprob <- min(1, a) #acceptance probability
    u <- runif(1, 0, 1)
    if(u < acprob) s[t] <- sprime else s[t] <- s[t-1]
  }
  return(s)
}

##-----------------------------------------------------------##

#Unconditional Score Vector
```

```
UCOND.S <- rep(0, 4)
#Unconditional Information Matrix
UCOND.I <- array(0, dim = c(4, 4))

R <- 10000
burn <- 1000

for(i in 1:N)
{
  COND.b <- rrandeff(R)
  ##-------------------------------------------------------##
  COND.S <- rep(0, 4)
  COND.I <- array(0, dim = c(4, 4))
  D <- array(0, dim = c(14, (R - burn)))
  for(j in 1:ncol(D))
  {
    D[,j] <- c(d.mu.1(x[i,], y[i,], COND.b[burn + j]),
               d.alpha.1(x[i,], y[i,],
                         COND.b[burn + j]),
               d.b.eta.1(ln.T[i,], ln.C[i,],
                        COND.b[burn + j], del[i,]),
               d.sig.b.1(COND.b[burn + j]),
               ##---------------------------------------##
               d.mu.2,
               d.mualpha.2(x[i,]),
               d.mub.eta.2,
               d.musig.b.2,
               d.alpha.2(x[i,]),
               d.alphab.eta.2,
               d.alphasig.b.2,
               d.b.eta.2(ln.C[i,], COND.b[burn + j],
                        del[i,]),
               d.b.etasig.b.2,
               d.sig.b.2(COND.b[burn + j])
    )
  }
  W <- apply(D, 1, mean)
  #Conditional Score Vector
  COND.S <- W[1:4]
  UCOND.S <- UCOND.S + COND.S

  ##-----------------------------------------------------##
```

```
    COND.I[1,1] <- W[5]
    COND.I[1,2] <- COND.I[2,1] <- W[6]
    COND.I[1,3] <- COND.I[3,1] <- W[7]
    COND.I[1,4] <- COND.I[4,1] <- W[8]
    COND.I[2,2] <- W[9]
    COND.I[2,3] <- COND.I[3,2] <- W[10]
    COND.I[2,4] <- COND.I[4,2] <- W[11]
    COND.I[3,3] <- W[12]
    COND.I[3,4] <- COND.I[4,3] <- W[13]
    COND.I[4,4] <- W[14] #Conditional Information Matrix
    ##-----------------------------------------------------##
    UCOND.I <- UCOND.I + COND.I
  }
  ucond <- list(score = UCOND.S, inf = -UCOND.I)
  return(ucond)
}

niter <- 80 # no. of iteration
res <- array(0, dim = c(niter,4))
colnames(res) <- c("mu", "alpha", "b.eta", "sig.b")

for(k in 1:niter)
{
  theta <- theta.ini + solve(Q(theta.ini)$inf) %*%
    Q(theta.ini)$score

  ## Updating estimates...
  mu <- theta[1]
  alpha <- theta[2]
  b.eta <- theta[3]
  sig.b <- theta[4]
  theta.ini <- c(mu, alpha, b.eta, sig.b)
  res[k,] <- theta.ini
  print(res[k,])
}

ul <- max(apply(res, 2, max)) # y-axis upper limit
ll <- min(apply(res, 2, min)) # y-axis lower limit

par(mar=par()$mar+c(0,0,0,3.5), xpd = T, cex = 0.95,
    family = "serif",
    font.axis = 3, font.lab = 3, font.main = 1)
```

```
plot(1:niter, res[,1], ylim = c(ll, ul),
     type = "l", lty = 1, lwd = 1.5,
     main = "Performance of the proposed algorithm",
     xlab = "Iteration Number",
     ylab = "Estimates")
lines(1:niter, res[,2], lty = 2, lwd = 1.5)
lines(1:niter, res[,3], lty = 3, lwd = 1.5)
lines(1:niter, res[,4], lty = 4, lwd = 1.5)
legend("bottomright", inset=c(-0.5,0),
       legend = c(expression(mu),
                  expression(alpha),
                  expression(beta),
                  expression(sigma[b])),
       lty = 1:4, lwd = rep(1.5,4), bty = "n")

## Standard error of the estimates
std.err <- apply(res, 2, sd)

##----------------------------END---------------------------##
```

## References


[1] Chakraborty, A. and Das, K. (2010). Inferences for joint modelling of repeated ordinal scores and time to event data. *Computational and Mathematical Methods in Medicine*, **11**, 281-295.

[2] DeGruttola, V. and Tu, X.M. (1964). Modelling progression of cd-4 lymphocyte count and its relationship to survival time, *Biometrics*, **50**, 1003-1014.

[3] Guo, X. and Carlin, B.P. (2004). Separate and joint modelling of longitudinal and event time data using standard computer packages. *The American Statistician*, **58**, 16-24.

[4] Henderson, R., Diggle, P.J. and Dobson, A. (2000). Joint modeling of longitudinal measurements and event time data, *Biometrics*, **1**, 465-80.

[5] Hogan, J. and Laird, N. (1997). Mixture models for the joint distribution of repeated measures and event times. *Statistics in Medicine*, **16**, 239-258.

[6] Huang, W., Zeger, S., Anthony, J. and Garrett, E. (2001). Latent variable model for joint analysis of multiple repeated measures and bivariate event times. *Journal of American Statistical Association*, **96(455)**, 906-914.

[7] Little, R. (1995). Modeling the dropout mechanism in repeatedmeasures studies. *Journal of the American Statistical Association*, **90**, 1112-1121.

[8] Pawitan, Y. and Self, S. (1993). Modeling disease marker processes in aids. *Journal of the American Statistical Association*, **88**, 719-726.

[9] Ratcliffe, S., Guo, G. and Ten Have, T. (2004). Joint modelling of longitudinal and survival data via a common frailty. *Biometrics*, **60(4)**, 892-899.

[10] R Core Team (2015). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

[11] Schluchter, M.D. (1992). Methods for the analysis of informatively censored longitudinal dada. *Statistics in Medicine*, **11**, 1861-1870.

[12] Ten Have, T., Miller, M., Reboussin, B. and James, M. (2000). Mixed effects logistic regression models for longitudinal ordinal functional response data with multiple-cause drop-out from longitudinal study of aging. *Biometrics*, **56**, 279-87.

[13] Wang, Y. and Taylor, J.M.G. (2001). Jointly modelling longitudinal and event time data with application to aids studies. *Journal of the American Statistical Association*, **96**, 895-905.

[14] Wulfson, M.S. and Tsiatis, A.A. (1997). A joint model for survival and longitudinal data measured with error. *Biometrics*, **53**, 330-339.

[15] Wu, M. and Carroll, R. (1988). Estimation and comparison of changes in the presence of informative right censoring by modeling the censoring process. *Biometrics*, **44**, 175-188.

[16] Wu, M. and Bailey, K. (1989). Estimation and comparison of change in the presence of informative right censoring: Conditional liner model. *Biometrics*, **44**, 175-188.

[17] Xu, J. and Zerger, S.L. (2001). Joint analysis of longitudinal data comprising repeated measures and times to events. *Journal of the Royal Statistical Society, Series C: Applied Statistics*, **50(3)**, 375-387.