Bayesian Hierarchical Weibull Regression

JQT paper with 200 drivers - Model 3

Miao Cai[*]

2018-12-05

1 Weibull Distribution

The probability density function of a Weibull distribution is

$$f(t) = \begin{cases} \frac{\kappa}{\theta}\left(\frac{t}{\theta}\right)^{\kappa-1} e^{(t/\theta)^{\kappa}}, & t > 8 \\ 0, & t \leq 0 \end{cases}$$

Then the survival function is:

$$S(t) = P(T > t) = 1 - P(T \leq t) = e^{-(t/\theta)^{\kappa}}$$

The hazard function is the first order derivative of the survival function:

$$h(t) = \frac{dS(t)}{dt} = \frac{\kappa}{\theta}\left(\frac{t}{\theta}\right)^{\kappa-1}$$

2 Weibull regression

We assume $Y_{i,d(i),s(i)}$ is the time until the first critical event from the start of a trip, then we have

$$\begin{aligned} Y_{i,d(i),s(i)} &\sim \text{WEIBULL}(\kappa, \theta) \\ \theta &= \exp(\beta_{0,d(i)} + \beta_{1,d(i)} \cdot \text{CT}_i + \xi \cdot \mathbf{W} + \nu \cdot \mathbf{D_i}) \end{aligned}$$

Where the θ is the scale parameter and κ is the shape parameter. When $\kappa > 1$, the hazard of having critical event is increasing as cumulative driving time gets longer, which indicates fatigued driving. When $\kappa = 1$, the Weibull distribution becomes an exponential distribution with constant hazard $\frac{1}{\theta}$, which indicates no fatigue. When $0 < \kappa < 1$, the hazard of having critical events is decreasing with longer cumulative driving time, which indicates anti-fatigue, or burn-in.

3 Censoring in Stan

In essense, Stan is evaluating a log probability function (probability density function or probability mass function) for a given set of parameters; this function returns the log density of the posterior up to an additive constant.

There are two ways of dealing with censoring in stan:

- estimating censored values
- integrating out censored values

[*]Department of Epidemiology and Biostatistics, Saint Louis University. Email address miao.cai@slu.edu

Estimating censored values in Stan is to treat these censored values as missing data with some constraint (lower bound, upper bound or interval). These censored values will be declared as parameters with constraints, and then estimated or imputed by their distribution. However, this way is not feasible when the censored data are huge, which leads to a huge amount parameters to be estimated.

The other way to deal with censored data is integrating out the censored values. Let's assume the data are right censored, which is common in survival data analysis. The probability of each censored datum can be calculated as:

$$P(t > C) = \int_C^\infty f(t|\kappa, \theta)dt = 1 - \Lambda(C|\kappa, \theta)$$

Where $f(t|\kappa, \theta)$ and $\Lambda(t|\kappa, \theta)$ are the probability density function and cumulative density function of Weibull distribution.

The log of total probability of N censored points is:

$$\log \prod_{i=1}^{N} P(t_i > C_i) = \sum_{i=1}^{N} \log \Big(1 - \Lambda(C_i|\kappa, \theta)\Big) \tag{1}$$

Stan provides `<distribution>_lccdf` as the log of complementary CDF for built-in distributions, so `weibull_lccdf` is the log of complementary CDF of weibull distribution. The Equation @ref(ltotalpdf) can be expressed as the following function in Stan:

$$\log \prod_{i=1}^{N} P(t_i > C_i) = \sum_{i=1}^{N} \texttt{weibull\_lccdf}(C_i|\kappa, \theta)$$

When it is translated into Stan `model` block, it becomes:

```
for (i in 1:N){
  target += weibull_lccdf(censored_time[i] | kappa, theta)
}
```

4 Examples

This is the Mice example of Weibull regression, originally provided in WinBUGS code, and then rewrote by the Stan team on their Github example models.

4.1 Data

```
#rm(list=ls(all=TRUE))
require(rstan)
require(shinystan)

N_uncensored <- 65L
N_censored <- 15L
M <- 4
group_uncensored <-
c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
```

```
4, 4, 4)
group_censored <-
c(1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4)
t_uncensored <-
c(12, 17, 21, 25, 11, 26, 27, 30, 13, 12, 21, 20, 23, 25, 23,
29, 35, 31, 36, 32, 27, 23, 12, 18, 38, 29, 30, 32, 25, 30, 37,
27, 22, 26, 28, 19, 15, 12, 35, 35, 10, 22, 18, 12, 31, 24, 37,
29, 27, 18, 22, 13, 18, 29, 28, 16, 22, 26, 19, 17, 28, 26, 12,
17, 26)
censor_time <-
c(40, 40, 40, 40, 40, 40, 40, 40, 10, 24, 40, 40, 20, 29, 10)

dataList = list(
  N_censored = N_censored ,
  N_uncensored = N_uncensored , # BUGS does not treat 1-column mat as vector
  M = M ,
  group_uncensored = group_uncensored ,
  group_censored = group_censored,
  censor_time = censor_time,
  t_uncensored = t_uncensored
)
```

The survival time for four treatment groups were observed. We split the observed data into two parts:

- uncensored MICE survival data
- censored MICE survival data

```
require(magrittr)
library(kableExtra)

knitr::kable(list(data.frame(group_uncensored, t_uncensored),
                  data.frame(group_censored, censor_time)),
             align = "c",
             caption = c("uncensored MICE survival data",
                         "censored MICE survival data")) %>%
  kableExtra::kable_styling(bootstrap_options = "striped",
                            full_width = F, position = "left") %>%
  kableExtra::scroll_box(width = "110%", height = "300px")
```

4.2 Official code provide by the Stan team

```
mice1 = "
data {
  int<lower=0> N_uncensored;
  int<lower=0> N_censored;
  int<lower=0> M;
  int<lower=1,upper=M> group_uncensored[N_uncensored];
  int<lower=1,upper=M> group_censored[N_censored];
  real<lower=0> censor_time[N_censored];
  real<lower=0> t_uncensored[N_uncensored];
}

parameters {
```

Table 1: uncensored MICE survival data

Table 2: censored MICE survival data

group_uncensored	t_uncensored
1	12
1	17
1	21
1	25
1	11
1	26
1	27
1	30
1	13
1	12
1	21
1	20
1	23
1	25
1	23
1	29
1	35
1	31
1	36
2	32
2	27
2	23
2	12
2	18
2	38
2	29
2	30
2	32
2	25
2	30
2	37
2	27
3	22
3	26
3	28
3	19
3	15
3	12
3	35
3	35
3	10
3	22
3	18
3	12
3	31
3	24
3	37
3	29
4	27
4	18
4	22
4	13
4	18

group_censored	censor_time
1	40
2	40
2	40
2	40
2	40
2	40
2	40
2	40
2	40
3	10
3	24
3	40
3	40
4	20
4	29
4	10

```
  real<lower=0> r;
  real beta[M];
  real<lower=1> t2_censored[N_censored]; // t_censored / censor_time
}

model {
  r ~ exponential(0.001);
  beta ~ normal(0, 100);
  for (n in 1:N_uncensored) {
    t_uncensored[n] ~ weibull(r, exp(-beta[group_uncensored[n]] / r));
  }
  for (n in 1:N_censored) {
    t2_censored[n] ~ weibull(r, exp(-beta[group_censored[n]] / r) / censor_time[n]);
  }
}

generated quantities {
  real median[M];
  real pos_control;
  real test_sub;
  real veh_control;

  for (m in 1:M)
    median[m] <- pow(log(2) * exp(-beta[m]), 1/r);

  veh_control <- beta[2] - beta[1];
  test_sub    <- beta[3] - beta[1];
  pos_control <- beta[4] - beta[1];
}
"

miceHMM1 <- stan(model_code=mice1, data=dataList, seed = 47306, chains=1,
                    iter=1000,
                    warmup=500 ) # init=initsChains
```

Notice that there are estimates for parameters `t2_censored[1]` to `t2_censored[15]` since this official Stan code is treating the censored data as parameters. The number of parameters for these censored values equals the number of censored values.

```
knitr::kable(summary(miceHMM1)$summary,
             digits = 2,
             caption = 'Summary statistics by the official Stan team',
             align = 'c') %>%
  kableExtra::kable_styling(bootstrap_options =
                              c("striped", "hover", "condensed")) %>%
  kableExtra::scroll_box(width = "800px", height = "700px")
```

4.3 Code by Miao

```
miao = "
data {
  int<lower=0> N_uncensored;
  int<lower=0> N_censored;
```

Table 3: Summary statistics by the official Stan team

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
r	3.28	0.03	0.33	2.71	3.03	3.27	3.50	3.97	105.46	1.01
beta[1]	-10.86	0.11	1.15	-13.21	-11.60	-10.84	-10.04	-8.81	108.22	1.01
beta[2]	-12.03	0.12	1.22	-14.52	-12.84	-12.03	-11.13	-9.90	106.02	1.01
beta[3]	-11.20	0.12	1.18	-13.64	-11.95	-11.18	-10.29	-9.22	101.12	1.01
beta[4]	-10.47	0.10	1.08	-12.62	-11.13	-10.47	-9.70	-8.49	107.78	1.01
t2_censored[1]	1.07	0.00	0.07	1.00	1.02	1.05	1.10	1.25	355.27	1.00
t2_censored[2]	1.19	0.01	0.16	1.01	1.07	1.15	1.27	1.62	500.00	1.01
t2_censored[3]	1.20	0.01	0.17	1.00	1.06	1.16	1.29	1.61	389.84	1.00
t2_censored[4]	1.22	0.01	0.18	1.01	1.08	1.17	1.31	1.64	500.00	1.00
t2_censored[5]	1.20	0.01	0.18	1.01	1.07	1.15	1.28	1.70	500.00	1.00
t2_censored[6]	1.19	0.01	0.15	1.01	1.07	1.17	1.27	1.59	409.00	1.00
t2_censored[7]	1.20	0.01	0.18	1.01	1.07	1.15	1.28	1.71	500.00	1.00
t2_censored[8]	1.20	0.01	0.18	1.01	1.06	1.14	1.29	1.64	500.00	1.00
t2_censored[9]	2.80	0.05	0.90	1.27	2.14	2.79	3.38	4.77	315.86	1.00
t2_censored[10]	1.34	0.02	0.27	1.01	1.12	1.29	1.51	1.96	227.04	1.02
t2_censored[11]	1.11	0.00	0.10	1.00	1.03	1.08	1.16	1.36	413.91	1.00
t2_censored[12]	1.10	0.00	0.10	1.00	1.03	1.08	1.15	1.38	408.35	1.00
t2_censored[13]	1.35	0.01	0.26	1.01	1.14	1.31	1.50	1.96	500.00	1.01
t2_censored[14]	1.13	0.01	0.12	1.01	1.04	1.10	1.17	1.44	500.00	1.00
t2_censored[15]	2.34	0.03	0.67	1.26	1.83	2.27	2.80	3.61	379.86	1.00
median[1]	24.49	0.10	1.89	21.05	23.18	24.54	25.75	28.25	352.91	1.00
median[2]	34.99	0.13	2.92	29.81	33.04	34.83	36.96	41.06	500.00	1.00
median[3]	27.15	0.11	2.18	23.03	25.72	27.04	28.54	31.86	415.10	1.01
median[4]	21.74	0.08	1.69	18.91	20.58	21.58	22.78	25.19	478.79	1.00
pos_control	0.39	0.02	0.34	-0.28	0.16	0.40	0.62	1.08	500.00	1.00
test_sub	-0.34	0.02	0.34	-1.02	-0.57	-0.32	-0.09	0.29	393.64	1.00
veh_control	-1.16	0.02	0.38	-1.94	-1.41	-1.18	-0.93	-0.42	500.00	1.00
lp___	-266.45	0.29	3.54	-273.89	-268.86	-266.23	-263.87	-260.54	147.76	1.02

```
  int<lower=0> M;
  int<lower=1,upper=M> group_uncensored[N_uncensored];
  int<lower=1,upper=M> group_censored[N_censored];
  real<lower=0> censor_time[N_censored];
  real<lower=0> t_uncensored[N_uncensored];
}

parameters{
  real<lower=0> r;
  vector[M] beta;
}

model{
  for (i in 1:N_uncensored){
    t_uncensored[i] ~ weibull(r, exp(-beta[group_uncensored[i]] / r));
  }
  for (j in 1:N_censored){
    target += weibull_lccdf(censor_time[j]|r, exp(-beta[group_censored[j]] / r));
  }

  r ~ gamma(1, 1);
  beta ~ normal(0, 100);

}

generated quantities {
  real median[M];
  real pos_control;
  real test_sub;
  real veh_control;

  for (m in 1:M)
    median[m] <- pow(log(2) * exp(-beta[m]), 1/r);

  veh_control <- beta[2] - beta[1];
  test_sub    <- beta[3] - beta[1];
  pos_control <- beta[4] - beta[1];
}
"


miceHMM2 <- stan(model_code=miao, data=dataList, seed = 47306, chains=1,
                 iter=1000,
                 warmup=500 ) # init=initsChains
```

Notice in the Stan results here, there are not posterior distribution estimates for `t2_censored[1]` to `t2_censored[15]`. This is because we are integrating out these censored data, instead of treating them as parameters.

```
knitr::kable(summary(miceHMM2)$summary,
             digits = 2,
             caption = 'Summary statistics by Miao',
             align = 'c') %>%
  kableExtra::kable_styling(bootstrap_options =
```

Table 4: Summary statistics by Miao

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
r	3.19	0.03	0.30	2.62	2.97	3.20	3.38	3.75	102.55	1.01
beta[1]	-10.54	0.10	1.03	-12.54	-11.23	-10.59	-9.79	-8.53	104.10	1.01
beta[2]	-11.69	0.11	1.11	-13.74	-12.42	-11.79	-10.85	-9.70	104.83	1.02
beta[3]	-10.88	0.10	1.04	-12.88	-11.61	-10.89	-10.16	-8.87	102.87	1.02
beta[4]	-10.16	0.10	0.99	-12.07	-10.83	-10.19	-9.44	-8.26	103.89	1.01
median[1]	24.30	0.08	1.75	21.10	23.18	24.18	25.42	27.83	500.00	1.00
median[2]	35.02	0.15	3.32	29.21	33.00	34.73	36.86	41.96	500.00	1.00
median[3]	27.08	0.10	2.32	22.61	25.58	27.00	28.40	32.17	500.00	1.01
median[4]	21.58	0.07	1.66	18.38	20.43	21.51	22.70	25.28	500.00	1.00
pos_control	0.38	0.01	0.32	-0.26	0.17	0.39	0.59	1.01	500.00	1.00
test_sub	-0.34	0.01	0.33	-1.02	-0.56	-0.34	-0.12	0.30	500.00	1.00
veh_control	-1.15	0.02	0.38	-1.96	-1.41	-1.14	-0.92	-0.40	500.00	1.00
lp___	-249.29	0.12	1.55	-253.18	-250.20	-248.95	-248.09	-247.20	177.41	1.01

```
                              c("striped", "hover", "condensed")) %>%
  kableExtra::scroll_box(width = "800px")
```

```
miao1 = "
data {
  int<lower=0> N_uncensored;
  int<lower=0> N_censored;
  int<lower=0> M;
  int<lower=1,upper=M> group_uncensored[N_uncensored];
  int<lower=1,upper=M> group_censored[N_censored];
  real<lower=0> censor_time[N_censored];
  real<lower=0> t_uncensored[N_uncensored];
}

parameters{
  real<lower=0> r;
  vector[M] beta;
}

model{
  for (i in 1:N_uncensored){
    target += weibull_lpdf(t_uncensored[i]|r, exp(-beta[group_uncensored[i]] / r));
  }
  for (j in 1:N_censored){
    target += weibull_lccdf(censor_time[j]|r, exp(-beta[group_censored[j]] / r));
  }

  r ~ gamma(1, 1);
  beta ~ normal(0, 100);

}

generated quantities {
  real median[M];
  real pos_control;
```

Table 5: Summary statistics by Miao

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
r	3.19	0.03	0.30	2.62	2.97	3.20	3.38	3.75	102.55	1.01
beta[1]	-10.54	0.10	1.03	-12.54	-11.23	-10.59	-9.79	-8.53	104.10	1.01
beta[2]	-11.69	0.11	1.11	-13.74	-12.42	-11.79	-10.85	-9.70	104.83	1.02
beta[3]	-10.88	0.10	1.04	-12.88	-11.61	-10.89	-10.16	-8.87	102.87	1.02
beta[4]	-10.16	0.10	0.99	-12.07	-10.83	-10.19	-9.44	-8.26	103.89	1.01
median[1]	24.30	0.08	1.75	21.10	23.18	24.18	25.42	27.83	500.00	1.00
median[2]	35.02	0.15	3.32	29.21	33.00	34.73	36.86	41.96	500.00	1.00
median[3]	27.08	0.10	2.32	22.61	25.58	27.00	28.40	32.17	500.00	1.01
median[4]	21.58	0.07	1.66	18.38	20.43	21.51	22.70	25.28	500.00	1.00
pos_control	0.38	0.01	0.32	-0.26	0.17	0.39	0.59	1.01	500.00	1.00
test_sub	-0.34	0.01	0.33	-1.02	-0.56	-0.34	-0.12	0.30	500.00	1.00
veh_control	-1.15	0.02	0.38	-1.96	-1.41	-1.14	-0.92	-0.40	500.00	1.00
lp___	-249.29	0.12	1.55	-253.18	-250.20	-248.95	-248.09	-247.20	177.41	1.01

```
  real test_sub;
  real veh_control;

  for (m in 1:M)
    median[m] <- pow(log(2) * exp(-beta[m]), 1/r);

  veh_control <- beta[2] - beta[1];
  test_sub    <- beta[3] - beta[1];
  pos_control <- beta[4] - beta[1];
}
"


miceHMM3 <- stan(model_code=miao1, data=dataList, seed = 47306, chains=1,
                 iter=1000,
                 warmup=500 ) # init=initsChains
```

```
knitr::kable(summary(miceHMM3)$summary,
             digits = 2,
             caption = 'Summary statistics by Miao',
             align = 'c') %>%
  kableExtra::kable_styling(bootstrap_options =
                              c("striped", "hover", "condensed")) %>%
  kableExtra::scroll_box(width = "800px")
```