A basic semicompeting risks model in Stan

Leah Comment

10/25/2017 (updated 10/28/2017)

Illness-death models

Introduction

The compartmental model representation of the issue of semicompeting risks conceptualizes an observational unit as belonging to exactly one state at any given time point. These models apply to contexts outside of illness and death, but it is illustrative to think of a person who may be "healthy," "ill," or "dead." Developing illness is a non-terminal event and dying is the terminal event. When illness occurs, it *must* occur prior to death. However, some subjects may die without ever experiencing the illness of interest.

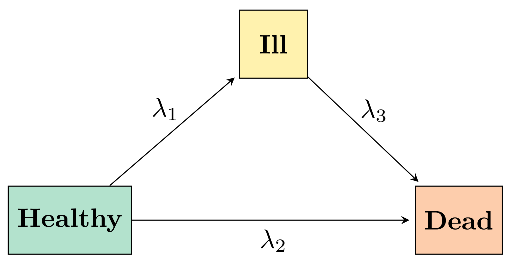


Figure 1: A compartmental model representation of an illness-death model in semicompeting risks

Making inferences in the illness-death context involves modeling the compartment transition hazards λ_1, λ_2, and vλ_3. These transitions hazards can change as a function of time, yielding $\lambda_1(t)$, $\lambda_2(t)$, and $\lambda_3(t)$.

Notation

Suppose we have a non-terminal event R and a terminal event T. The terminal event can be any absorbing state, but "terminal event" and "death" are used interchangeably here. Both event times are subject to a censoring time which may vary by person. Let T_i denote the terminal event time for person i. When the non-terminal event occurs, $R_i \in \mathbb{R}_+$; when the non-terminal event is truncated by death, $R_i \equiv \bar{\mathbb{R}}$, a non-real number. The potential censoring time is C_i. The observed non-terminal event time is $Y_i^R = \min(R_i, T_i, C_i)$, with $\min(x, \bar{\mathbb{R}}) \equiv x$ for any $x \in \mathbb{R}$. The non-terminal event indicator δ_i^R is $\mathbf{1}(Y_i^R = R_i)$. Similarly, $Y_i^T = \min(T_i, C_i)$, and δ_i^R is $\mathbf{1}(Y_i^T = T_i)$.

Potential observed data patterns

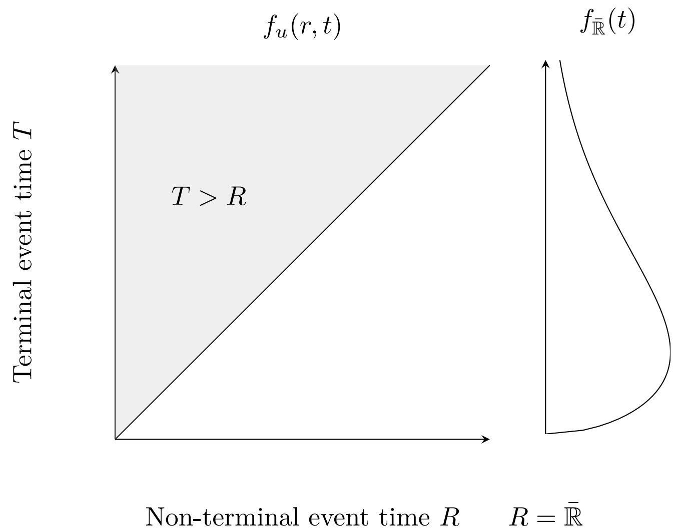
Defined by $V_i = (\delta_i^R, \delta_i^T)$, four distinct data patterns are possible:

Type	Description	(Y_i^R, Y_i^T)	$V_i = (\delta_i^R, \delta_i^T)$
1	Observe neither non-terminal nor terminal event	(c_i, c_i)	$(0, 0)$
2	Observe non-terminal event; terminal event is censored	$(r_i, c_i) = (y_i^R, c_i)$	$(1, 0)$

Type	Description	(Y_i^R, Y_i^T)	$V_i = (\delta_i^R, \delta_i^T)$
3	Observe terminal event without non-terminal event	$(t_i, c_i) = (y_i^T, y_i^T)$	$(0,1)$
4	Observe non-terminal and terminal events	$(r_i, t_i) = (y_i^R, y_i^T)$	$(1,1)$

Defining the joint distribution of non-terminal and terminal event times

The joint density for T and R is shown in Figure . The portion where both event types are observed is only defined on the "upper wedge" where $T > R$; the joint density is $f_u(r,t)$. When the non-terminal event does not occur ($R = \bar{\mathrm{R}}$), the density of T is give $f_{\bar{\mathrm{R}}}(t)$.



For now, we assume a semi-Markov model for λ_3 where the hazard of the terminal event only depends on covariates and time elapsed since the non-terminal event. Ignoring measure theoretic issues and using $P(X = x)$ to refer to the pdf $f(x)$ for any continuous random variable,

$$
\begin{aligned}
f_u(r,t) &= P(R = r, T = t) \text{ for } t > r \\
&= P(R = r, T > r) P(T = t | R = r, T > r) \\
&= \underbrace{\frac{P(R = r, T > r)}{P(R > r, T > r)}}_{\lambda_1(r)} \underbrace{P(R > r, T > r)}_{S_1(r)S_2(r)} \underbrace{\frac{P(T = t | R = r, T > r)}{P(T > t | R = r)}}_{\lambda_3(t-r)} \underbrace{P(T > t | R = r)}_{S_3(t-r)} \\
&= \lambda_1(r) \lambda_3(t-r) \exp\{-\Lambda_1(r) - \Lambda_3(t-r)\}
\end{aligned}
$$

In a Markov model for death following the non-terminal event, $\lambda_3(t-r)$, $\Lambda_3(t-r)$, and $S_3(t-r)$ above would be replaced by $\lambda_3(t)$, $\Lambda_3(t)$, and $S_3(t)$.

The density for terminal event times among those who never experience the non-terminal event is

$$
\begin{aligned}
f_{\bar{\mathbb{R}}} &= P(T = t, R = \bar{\mathbb{R}}) \\
&= \underbrace{\frac{P(R > t, T = t)}{P(R > r, T > r)}}_{\lambda_2(t)} \underbrace{P(R > r, T > r)}_{S_1(t)S_2(t)} \\
&= \lambda_2(t) \exp\{-\Lambda_1(t) + \Lambda_2(t)\}
\end{aligned}
$$

Likelihood contributions by data pattern

For the semi-Markov model, the general form of the likelihood contributions for each observed data pattern in the table above is

$$
(\text{non-terminal contribution}) \times (\text{terminal contribution})
$$

Type 1 (observe neither event type):

$$
L_{1i} = P(R > c_i, T > c_i) = S_1(c_i)S_2(c_i) = S_1(y_i^R)S_2(y_i^T)
$$

Type 2 (observe only non-terminal event):

$$
L_{2i} = P(R = y_i^R, T > c_i) = \int_{c_i}^{\infty} f_u(y_i^R, u)du = f_1(y_i^R)S_2(y_i^R)S_3(c_i - y_i^R) = f_1(y_i^R)S_2(y_i^R)S_3(y_i^T - y_i^R)
$$

Type 3 (observe only terminal event):

$$
L_{3i} = P(R > y_i^T, T = y_i^T) = \int_{y_i^T}^{\infty} f_u(u, y_i^R)du = S_1(y_i^T)f_2(y_i^T)
$$

Type 4 (observe both event types):

$$
L_{4i} = P(R = y_i^R, T = y_i^T) = f_u(y_i^R, y_i^T) = f_1(y_i^R)S_2(y_i^R)f_3(y_i^T - y_i^R)
$$

Full likelihood

The combined likelihood of the observed data for n observations is

$$
\begin{aligned}
\mathcal{L}_n = &\left[\prod_{i:(\delta_i^R, \delta_i^T)=(0,0)} S_1(y_i^R)S_2(y_i^T)\right] \left[\prod_{i:(\delta_i^R, \delta_i^T)=(1,0)} f_1(y_i^R)S_2(y_i^R)S_3(y_i^T - y_i^R)\right] \times \\
&\left[\prod_{i:(\delta_i^R, \delta_i^T)=(0,1)} S_1(y_i^T)f_2(y_i^T)\right] \left[\prod_{i:(\delta_i^R, \delta_i^T)=(1,1)} f_1(y_i^R)S_2(y_i^R)f_3(y_i^T - y_i^R)\right]
\end{aligned}
$$

TODO(LCOMM): fill in notation gaps above

Model formulation

Weibull models

The hazard of a Weibull regression model with shape α is

$$h(t|x_i) = \alpha t^{\alpha-1}\exp\{\beta_0 + x_i'\beta\}$$

With $\kappa = \exp(\beta_0)$, this is the parameterization used in Lee et al.

Alternatively, one can use the Stan hazard parameterization, where

$$h(t) = \frac{\alpha}{\sigma}\left(\frac{t}{\sigma}\right)^{\alpha-1}$$

To incorporate regression parameters into the model, we note that

$$\sigma^{-\alpha} = \exp\{x_i'\beta\}$$

and thus

$$\sigma = \exp\left\{-\frac{x_i'\beta}{\alpha}\right\}$$

Fitting semicompeting risk Weibull hazard models in Stan

Suppose we use Weibull hazards for the illness-death model below, which includes only a single covariate (`x_c` in the `dat_ID` data set):

Non-terminal event hazard at time r:

$$h_1(r|X_i) = \alpha_1\kappa_1 r^{\alpha_1-1}\exp\{\beta_1 X_i\} = \alpha_1 r^{\alpha_1-1}\exp\{\beta_0 + \beta_1 X_i\} \text{ for } r > 0$$

Terminal event hazard at time t, without having experienced the non-terminal event:

$$h_2(t|X_i) = \alpha_2\kappa_2 t^{\alpha_2-1}\exp\{\beta_2 X_i\} = \alpha_2 t^{\alpha_2-1}\exp\{\beta_{02} + \beta_2 X_i\} \text{ for } t > 0$$

Terminal event hazard at time t, after having experienced the non-terminal event at r in a Semi-Markov model:

$$h_3(t|X_i) = \alpha_3\kappa_3(t-r)^{\alpha_3-1}\exp\{\beta_3 X_i\} = \alpha_3(t-r)^{\alpha_3-1}\exp\{\beta_{03} + \beta_3 X_i\} \text{ for } t > r$$

Let $\mathbf{x_i'}\boldsymbol{\beta}_g = \beta_{0g} + \beta_g X_i$ for $g \in (1,2,3)$. (This is the linear predictor that appears in the Stan code.)

```
# Packages
library("knitr")
suppressPackageStartupMessages(library("rstan"))
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores() - 1)
```

```
# Read in semicompeting risk Weibull simulated data
dat_ID <- readRDS("dat_ID.Rdata")

# Package for Stan
x_m <- cbind(1, dat_ID$x_c)
stan_dat <-  list(N = nrow(dat_ID),
                  P_1 = ncol(x_m),
```

```
                   X_1 = x_m,
                   P_2 = ncol(x_m),
                   X_2 = x_m,
                   P_3 = ncol(x_m),
                   X_3 = x_m,
                   Yr = dat_ID$R,
                   dYr = dat_ID$delta_R,
                   Yt = dat_ID$T,
                   dYt = dat_ID$delta_T)
```

The Stan code to fit these models is shown below.

```
data {
  // number of observations
  int<lower=0> N;
  // number of columns in 1st design matrix, including intercept
  int<lower=1> P_1;
  // design matrix for non-terminal model
  matrix[N, P_1] X_1;
  // number of columns in 2nd design matrix, including intercept
  int<lower=1> P_2;
  // design matrix for terminal model w/o non-terminal history
  matrix[N, P_2] X_2;
  // number of columns in 3rd design matrix, including intercept
  int<lower=1> P_3;
  // design matrix for terminal model with non-terminal history
  matrix[N, P_3] X_3;
  // observed non-terminal time
  vector<lower=0>[N] Yr;
  // indicator of event observation for non-terminal event
  int<lower=0,upper=1> dYr[N];
  // observed terminal time
  vector<lower=0>[N] Yt;
  // indicator of event observation for terminal event
  int<lower=0,upper=1> dYt[N];
}

transformed data {
  // Duration of gap between non-terminal and terminal events
  vector[N] YtYrdiff;
  YtYrdiff = Yt - Yr;
}

parameters {
  // vectors of regression parameters
  vector[P_1] beta1;
  vector[P_2] beta2;
  vector[P_3] beta3;

  // shape parameters (the one in exponent of time)
  // alpha > 1 -> hazard increases over time, more clumping
  real<lower=0> alpha1;
  real<lower=0> alpha2;
  real<lower=0> alpha3;
```

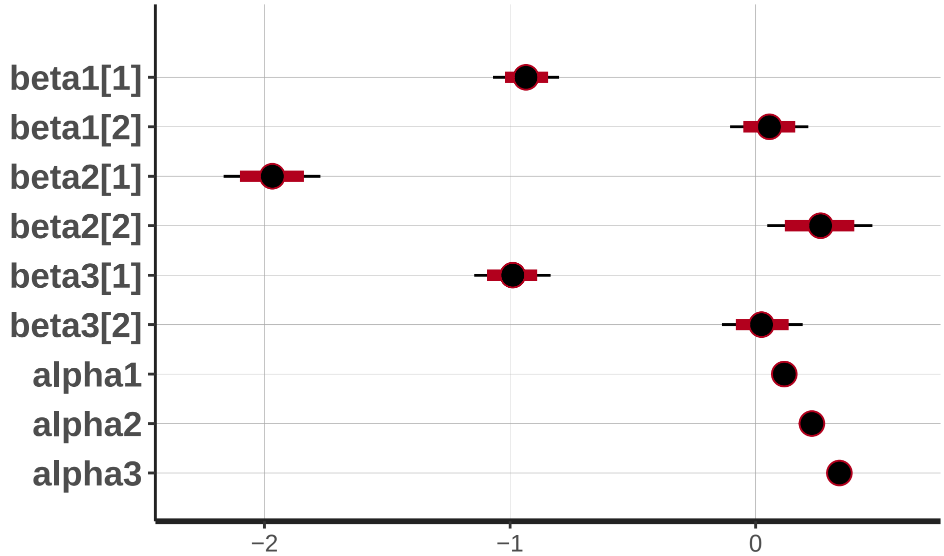
```
  // scale parameters
  // bigger sigma -> slower event occurrence (double check this)
  //real<lower=0> sigma1;
  //real<lower=0> sigma2;
  //real<lower=0> sigma3;
}

model {
  // linear predictors
  vector[N] lp1;
  vector[N] lp2;
  vector[N] lp3;
  lp1 = X_1 * beta1;
  lp2 = X_2 * beta2;
  lp3 = X_3 * beta3;

  // no priors -> use Stan defaults

  // likelihood
  for (n in 1:N){
    if (dYr[n] == 0 && dYt[n] == 0) {

      // type 1: observe neither event
      target += weibull_lccdf(Yr[n] | alpha1, exp(-(lp1[n])/alpha1)) +
                weibull_lccdf(Yt[n] | alpha2, exp(-(lp2[n])/alpha2));

    } else if (dYr[n] == 1 && dYt[n] == 0) {

      // type 2: observe non-terminal but terminal censored
      target += weibull_lpdf(Yr[n] | alpha1, exp(-(lp1[n])/alpha1)) +
                weibull_lccdf(Yr[n] | alpha2, exp(-(lp2[n])/alpha2)) +
                weibull_lccdf(YtYrdiff[n] | alpha3, exp(-(lp3[n])/alpha3));

    } else if (dYr[n] == 0 && dYt[n] == 1) {

      // type 3: observed terminal with no prior non-terminal
      target += weibull_lccdf(Yr[n] | alpha1, exp(-(lp1[n])/alpha1)) +
                weibull_lpdf(Yt[n] | alpha2, exp(-(lp2[n])/alpha2));

    } else if (dYr[n] == 1 && dYt[n] == 1) {

      // type 4: both non-terminal and terminal observed
      target += weibull_lpdf(Yr[n] | alpha1, exp(-(lp1[n])/alpha1)) +
                weibull_lccdf(Yr[n] | alpha2, exp(-(lp2[n])/alpha2)) +
                weibull_lpdf(YtYrdiff[n] | alpha3, exp(-(lp3[n])/alpha3));

    }
  }
}
```

```
# Fit Weibull semicompeting model
library("rstan")
# Discards first half of chains -> will yield 2000 posterior samples total
```

```r
fit1 <- stan(file = "semicompeting_weibull.stan", data = stan_dat,
             iter = 1000, chains = 4)
s_elapsed <- sum(get_elapsed_time(fit1))
plot(fit1)
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```




```r
print(fit1)
```

```
## Inference for Stan model: semicompeting_weibull.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##               mean se_mean   sd     2.5%      25%      50%      75%
## beta1[1]     -0.93    0.00 0.07    -1.07    -0.98    -0.94    -0.89
## beta1[2]      0.06    0.00 0.08    -0.10     0.00     0.06     0.11
## beta2[1]     -1.97    0.00 0.10    -2.17    -2.04    -1.97    -1.90
## beta2[2]      0.26    0.00 0.11     0.05     0.19     0.26     0.34
## beta3[1]     -0.99    0.00 0.08    -1.15    -1.04    -0.99    -0.93
## beta3[2]      0.03    0.00 0.08    -0.14    -0.03     0.02     0.08
## alpha1        0.12    0.00 0.00     0.11     0.11     0.12     0.12
## alpha2        0.23    0.00 0.01     0.21     0.22     0.23     0.24
## alpha3        0.34    0.00 0.01     0.32     0.33     0.34     0.35
## lp__      -3214.24    0.07 1.98 -3218.78 -3215.35 -3213.95 -3212.80
##              97.5% n_eff Rhat
## beta1[1]     -0.80  2000    1
## beta1[2]      0.21  2000    1
## beta2[1]     -1.77  2000    1
## beta2[2]      0.48  2000    1
## beta3[1]     -0.83  2000    1
## beta3[2]      0.19  2000    1
## alpha1        0.12  2000    1
## alpha2        0.25  2000    1
## alpha3        0.36  2000    1
## lp__      -3211.29   865    1
```

```
##
## Samples were drawn using NUTS(diag_e) at Sat Oct 28 13:17:52 2017.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

TODO(LCOMM): Look into reparameterization to reduce numerical issues

We use the `SemiCompRisks` package to fit the illness-death model. The model fit by `BayesID_HReg()` is not completely comparable to the Stan code above because it includes an individual-level γ frailty that induces additional correlation between R_i and T_i. (However, the Stan code could be changed to include a frailty parameter.)

The models fit by `SemiCompRisks` are as follows:

Non-terminal event hazard at time r:

$$h_1(r|\gamma_i, X_i) = \gamma_i \alpha_1 \kappa_1 r^{\alpha_1 - 1} \exp\{\beta_1 X_i\} = \gamma_i \alpha_1 r^{\alpha_1 - 1} \exp\{\beta_0 + \beta_1 X_i\} \text{ for } r > 0$$

Terminal event hazard at time t, without having experienced the non-terminal event:

$$h_2(t|\gamma_i, X_i) = \gamma_i \alpha_2 \kappa_2 t^{\alpha_2 - 1} \exp\{\beta_2 X_i\} = \gamma_i \alpha_2 t^{\alpha_2 - 1} \exp\{\beta_{02} + \beta_2 X_i\} \text{ for } t > 0$$

Terminal event hazard at time t, after having experienced the non-terminal event at r in a Semi-Markov model:

$$h_3(t|\gamma_i, X_i) = \gamma_i \alpha_3 \kappa_3 (t-r)^{\alpha_3 - 1} \exp\{\beta_3 X_i\} = \gamma_i \alpha_3 (t-r)^{\alpha_3 - 1} \exp\{\beta_{03} + \beta_3 X_i\} \text{ for } t > r$$

The γ_i are assumed to be i.i.d. from a $\Gamma(\theta^{-1}, \theta^{-1})$ distribution, which has a mean of 1. The models assumed by the Stan code can be seen as a special case where $\gamma_i = 1 \; \forall \; i$.

```r
# Fit SemiCompRisks models
k_start <- Sys.time()
fit2 <- BayesID_HReg(Y, lin.pred, data = dat_ID,
                     hyperParams = hyperParams, startValues = startValues,
                     mcmc = mcmc.WB, path = "SemiCompRisks/Output/")
```

```
## chain:  1
## iteration: 10000: Sat Oct 28 13:32:14 2017
##
## chain:  2
## iteration: 10000: Sat Oct 28 13:32:17 2017
##
## chain:  3
## iteration: 10000: Sat Oct 28 13:32:19 2017
##
## chain:  4
## iteration: 10000: Sat Oct 28 13:32:21 2017
```

```r
k_elapsed <- Sys.time() - k_start
```

The Stan model took 41.4 seconds and the `SemiCompRisks` model took 8.44 seconds to run. Although the models are not exactly comparable, we can compare our estimates from these two fits, especially for $(\beta_1, \beta_2, \beta_3)$.

TODO(LCOMM): Check on these times (expes) TODO(LCOMM): Get effective sample size/PSRF numbers to compare 2 approaches.

```r
# Comparing estimates from SemiCompRisks and Stan models

# SemiCompRisks posterior means? medians? can't tell
# TODO(LCOMM): Find this out
k_theta_est <- summary(fit2)[["theta"]][,1]
k_kappa_ests <- exp(summary(fit2)[["h0"]][1, c(1,4,7)])
k_beta0_ests <- summary(fit2)[["h0"]][1, c(1,4,7)]
k_alpha_ests <- exp(summary(fit2)[["h0"]][2, c(1,4,7)])
k_beta1_ests <- log(summary(fit2)[["coef"]][c(1,4,7)])
semicomp_param_ests <- c(k_theta_est, k_beta0_ests, k_beta1_ests, k_alpha_ests)

# Stan posterior means
s_pmeans <- summary(fit1)[["summary"]][,"mean"]
s_beta0_ests <- s_pmeans[c("beta1[1]", "beta2[1]", "beta3[1]")]
s_beta1_ests <- s_pmeans[c("beta1[2]", "beta2[2]", "beta3[2]")]
s_alpha_ests <- s_pmeans[c("alpha1", "alpha2", "alpha3")]
names(s_beta0_ests) <- c("$\\beta_{01}$", "$\\beta_{02}$", "$\\beta_{03}$")
names(s_beta1_ests) <- c("$\\beta_{1}$", "$\\beta_{2}$", "$\\beta_{3}$")
names(s_alpha_ests) <- c("$\\alpha_{1}$", "$\\alpha_{2}$", "$\\alpha_{3}$")
stan_param_ests <- c(s_beta0_ests, s_beta1_ests, s_alpha_ests)

# Combine into a table
library("knitr")
semi_res_tab <- data.frame(Parameter = c("$\\theta$", names(stan_param_ests)),
                           stan = c(NA, stan_param_ests),
                           semicomp = semicomp_param_ests)
options(knitr.kable.NA = "--")
kable(semi_res_tab, escape = FALSE, row.names = FALSE, align = "crr",
      col.names = c("Parameter", "Estimate from Stan", "Estimate from SemiCompRisks"),
      caption = "Illness-death parameter estimates", digits = 4)
```

Table 2: Illness-death parameter estimates

Parameter	Estimate from Stan	Estimate from SemiCompRisks
θ	–	0.1901
β_{01}	-0.9346	-2.0969
β_{02}	-1.9689	-1.4082
β_{03}	-0.9898	-1.0122
β_1	0.0565	0.0732
β_2	0.2625	0.2692
β_3	0.0261	0.0221
α_1	0.1168	0.4123
α_2	0.2293	0.1487
α_3	0.3413	0.3536

TODO(LCOMM): Make posterior density plot comparison (maybe?)

TODO(LCOMM): Some kind of convergence comparison -> PSRF?

TODO(LCOMM): Implement Stan model with frailty