blupsurv Package Example (Version 0.1-7)

Emmanuel Sharef

April 21, 2008

1 Introduction

Here we demonstrate the use of the `blupsurv` package for analyzing a clustered bivariate data set using the `bivrec` method. Because we are not aware of any real-world data sets that can be freely distributed, we use a simulated data set for illustration.

In section 2, we give a short overview of the model structure. Section 3 demonstrates the analysis of the example data set.

2 Model overview

The model assumes that observed data consists of m independent clusters of J_i subjects, $i = 1, \ldots, m$. Each subject (i, j), experiences $N_{ij}^{(d)}$ observed recurrent events of type $d \in \{0, 1\}$, occurring at times $0 < S_{ij1}^{(d)} < \ldots < S_{ijN_{ij}^{(d)}}^{(d)}$, prior to a censoring time C_{ij}. Denote the recurrent event counting process for each subject as $N_{ij}^{(d)}(t)$, so that $N_{ij}^{(d)} = N_{ij}^{(d)}(C_{ij})$. Subjects may have time-dependent covariates $Z_{ij}(t)$ and may be stratified into p levels by a stratum indicator $L_{ij}(t)$.

Correlations between subjects within the same cluster and between event times are captured by nested frailties. Specifically, cluster frailties for each event type $U_*^{(d)} = \left(U_1^{(d)}, \ldots, U_m^{(d)}\right)$ are assumed to be positive and independent, with

$$\mathbb{E}\left[U_i^{(d)}\right] = 1 \,, \qquad \mathrm{Var}\left(U_i^{(d)}\right) = \sigma_{(d)}^2 \,. \tag{1}$$

Subject-level frailties are assumed to be positive and independent conditional on the cluster-level frailties, with

$$\mathbb{E}\left[U_{ij}^{(d)} | U_*^{(d)} = u_*^{(d)}\right] = u_i^{(d)} \tag{2}$$

$$\mathrm{Var}\left(U_{ij}^{(d)} | U_*^{(d)} = u_*^{(d)}\right) = u_i \nu_{(d)}^2 \tag{3}$$

$$\mathrm{Cov}\left(U_{ij}^{(0)}, U_{ij}^{(1)} | U_*^{(*)} = u_*^{(*)}\right) = \theta \,, \tag{4}$$

for all $i = 1, \ldots, m, j = 1, \ldots, J_i$.

Note that this implies that the marginal correlation between frailties for the two recurrent event types is given by

$$\rho = \mathrm{Cor}(U_{ij}^{(0)}, U_{ij}^{(1)}) = \theta \prod_{d \in \{0,1\}} \left(\sigma_{(d)}^2 + \nu_{(d)}^2\right)^{-\frac{1}{2}} \,. \tag{5}$$

Conditional on the frailties, the intensities for the recurrent event processes may be modeled as

$$\lambda_{ij}^{(d)}(t) = \lambda_{0L_{ij}(t)}^{(d)}\left(t - S_{ijN_{ij}^{(d)}(t)}^{(d)}\right) \cdot U_{ij}^{(d)} e^{\beta^{(d)} Z_{ij}(t)} \,, \tag{6}$$

where $\beta^{(d)}$ are regression coefficients, $\lambda_{0r}^{(d)}$ are stratum-specific baseline hazards for strata $r = 1, \ldots, p$, and $U_{ij}^{(d)}$ are frailties with the given moment structure. Denote the cumulative hazard as $\Lambda_{ij}^{(d)}(t) = \int_0^t \lambda_{ij}^{(d)}(u)du$. Note that the recurrent event intensity is specified in terms of the gap times.

In order to avoid numerical instabilities and allow for faster fitting, we allow a discretization to be imposed, by making the additional assumption that the baseline hazards for each stratum are constant and finite during $K_r^{(d)}$ time intervals ($r = 1, \ldots, p$). Breakpoints in the hazard may be chosen a priori, or based on observed data. Denote the breakpoints in the baseline hazards as $0 < a_{r1}^{(d)} < \ldots < a_{rK_r^{(d)}}^{(d)}$, so that the baseline hazards have value

$$\lambda_{0r}^{(d)}(t) = \sum_{s=1}^{K_r^{(d)}} \alpha_{rs}^{(d)} I(t \in [a_{rs-1}^{(d)}, a_{rs}^{(d)})) \,, \tag{7}$$

Estimation of regression and frailty dispersion parameters takes the form of a three-step Expectation-Maximization (EM) algorithm. Fixing the regression and dispersion parameters allows the frailties to be estimated by their orthodox best linear unbiased predictors under an auxiliary Poisson model. Updated dispersion parameters are then computed by bias-corrected Pearson estimators using the frailty estimates. Conditionally on the estimated frailties, the regression parameters can be obtained by maximizing the conditional profile likelihood.

For further detail on the model-fitting procedure, consult the accompanying technical report.

3 Example data analysis

We now analyze the included simulated data set `vigndata`. Begin by loading the package and data:

```
> library(blupsurv)
> data(vigndata)
```

As noted in the introduction, this is a simulated data set. However, for purposes of illustration, we have given the clusters and covariates real-world names. The data consist of 805 patients in 50 clusters corresponding to the states of the U.S., who were observed during a 10-year study for episodes of severe pain or fever. The following shows the first 20 rows of the data:

```
> vigndata[1:20, ]
```

	state	patientID	start	stop	pain	fever	age	sex
1	AL	1	0.000000	10.000000	0	0	51	M
2	AL	2	0.000000	10.000000	0	0	61	F
3	AL	3	0.000000	10.000000	0	0	52	F
4	AL	4	0.000000	10.000000	0	0	41	F
5	AL	5	0.000000	10.000000	0	0	49	M
6	AL	6	0.000000	10.000000	0	0	45	F
7	AL	7	0.000000	1.975463	1	0	51	M
8	AL	7	1.975463	3.295919	0	1	51	M
9	AL	7	3.295919	5.333440	1	0	51	M
10	AL	7	5.333440	10.000000	0	0	51	M
11	AL	8	0.000000	1.751679	0	1	55	M
12	AL	8	1.751679	10.000000	0	0	55	M
13	AL	9	0.000000	2.215776	1	0	43	M
14	AL	9	2.215776	4.190005	0	1	43	M
15	AL	9	4.190005	9.719571	1	0	43	M

```
16    AL         9 9.719571 10.000000    0     0  43   M
17    AL        10 0.000000 10.000000    0     0  46   F
18    AK         1 0.000000 10.000000    0     0  64   F
19    AK         2 0.000000  4.600518    1     0  66   F
20    AK         2 4.600518 10.000000    0     0  66   F
```

The data contains 664 pain events, and 493 fever events. We omit further descriptive analysis of the data, since it lies outside the scope of this package.

We can obtain a "quick" fit of the data by using a very coarse discretization, only allowing the baseline hazards to change 10 times during the study period[1]:

```
> vigndata.quickfit <- bivrec(Surv2(start, stop, pain, fever) ~ age +
+     sex + cluster(state) + id(patientID), data = vigndata, K1 = 10,
+     K2 = 10, verbose = 0)
```

Note the features of the above call. The response is a bivariate survival (`Surv2`) object, with event indicators `pain` and `fever`. In addition to the covariates `age` and `sex`, we also have to specify the cluster and subject identifiers, with the `cluster()` and `id()` terms. The options `K1` and `K2` control the discretization for the pain and fever hazards.

Here is a summary of the fit results:

```
> summary(vigndata.quickfit)

Summary of regression coefficients:
     pain.coef  pain.sd pain.pval     fever.coef fever.sd fever.pval
age   0.003912 0.003867    0.1559         0.0226 0.004708      <1e-4 ***
sexM  1.121538 0.087452     <1e-4 ***     1.6700 0.118246      <1e-4 ***

Summary of dispersion coefficients:

var.pain.clust  0.11776
var.fever.clust 0.16342
var.pain.subj   0.05348
var.fever.subj  0.11408
covariance      0.03451
correlation     0.15830
```

The summary shows that age is significant for fever, but not for pain, and that sex is significant for both processes. Furthermore, we see that both cluster- and subject-level frailty variances are not negligible, with variances for fever generally higher than those for pain. The two processes are also somewhat correlated.

Keep in mind that this fit was done with very coarse discretization, and simulation studies have shown that regression coefficients and frailty dispersion parameters are generally underestimated with coarse discretization, so the effect may be quite a bit larger!

Figure 1 shows the plot of survivor functions, produced by the following R statement:

```
> plot(vigndata.quickfit)
```

Note the evident discretization in the baseline hazard.

While the results using coarse discretizations may not be very precise, they are adequate for model selection. Based on the summary above, we conclude that excluding the age covariate for the fever process may be appropriate. We run another "quick" fit, applying the exclusion:

[1]This vignette "cheats" a little: since these fits are quite computationally expensive, their results are already included in the package, so they don't need to be run at build-time.

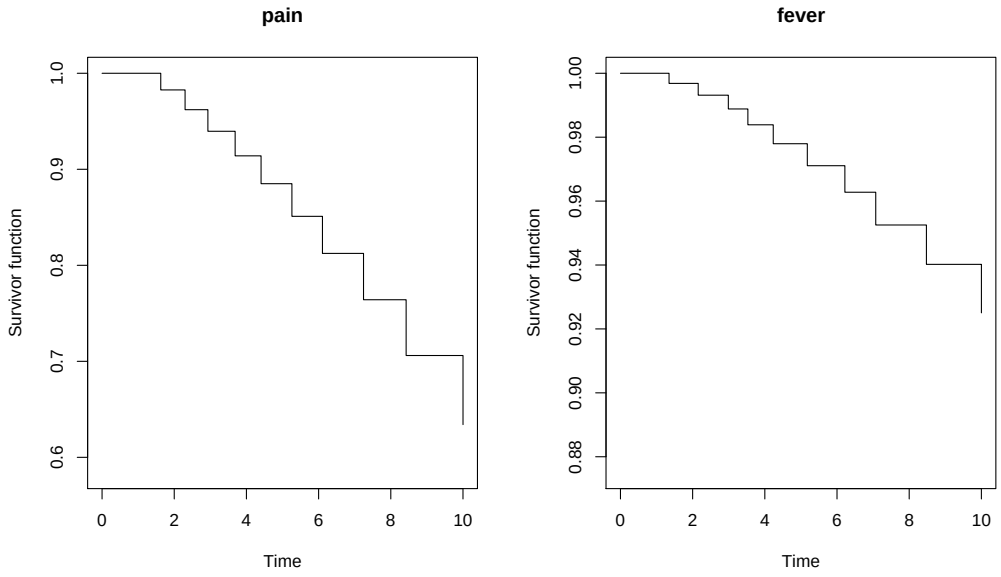

Figure 1: Plot of the "quick" fit results, with very coarse discretization.

```
> vigndata.quickfit2 <- bivrec(Surv2(start, stop, pain, fever) ~ age +
+     sex + cluster(state) + id(patientID), K1 = 10, K2 = 10, excludevars1 = "age",
+     data = vigndata, verbose = 0)

> summary(vigndata.quickfit2)

Summary of regression coefficients:
     pain.coef pain.sd pain.pval     fever.coef fever.sd fever.pval
sexM     1.118 0.08738     <1e-4 ***    1.79941 0.122731      <1e-4 ***
age                                     0.01804 0.004692      <1e-4 ***

Summary of dispersion coefficients:

var.pain.clust  0.11732
var.fever.clust 0.17001
var.pain.subj   0.05326
var.fever.subj  0.11275
covariance      0.03506
correlation     0.15963
```

The fitted values are quite similar, unsurprisingly. Lastly, we fit a model using the finest possible discretization. Rather than explicitly giving the number of desired breakpoints in the baseline hazard, setting `K1=1, K2=1` signals that the maximum number of breakpoints is desired:

```
> vigndata.fit <- bivrec(Surv2(start, stop, pain, fever) ~ age + sex +
+     cluster(state) + id(patientID), K1 = 1, K2 = 1, excludevars1 = "age",
+     data = vigndata, verbose = 0)

> summary(vigndata.fit)
```

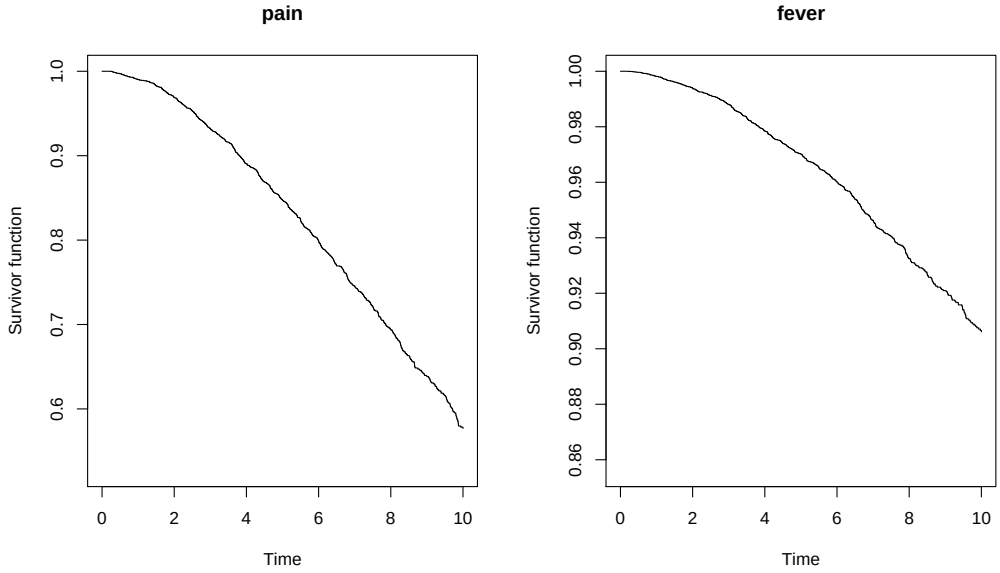

Figure 2: Plot of the final model fit results, with fine discretization.

```
Summary of regression coefficients:
     pain.coef pain.sd pain.pval     fever.coef fever.sd fever.pval
sexM     1.157 0.09206     <1e-4 ***    1.79941 0.126358      <1e-4 ***
age                                     0.01733 0.005113  0.0003496 ***

Summary of dispersion coefficients:

var.pain.clust  0.1787
var.fever.clust 0.2405
var.pain.subj   0.1633
var.fever.subj  0.3736
covariance      0.1547
correlation     0.3375
```

Note that almost all the parameter estimates are larger than the "quick" fit earlier, especially the estimates of the frailty dispersion parameters. Figure 2 shows the plot of survivor functions, produced by the following R statement:

```
> plot(vigndata.fit)
```

At fine levels of discretization, the survivor function is smooth, and in fact has the Weibull shape used in generating the simulated data. We can also examine the estimated frailty structure. Figure 3 shows a boxplot of the estimated subject-level frailties in each cluster:

```
> Ji <- table(substr(names(vigndata.fit$frailty$subj1), 1, 2))
> groups = rep(names(Ji), Ji)
> par(mfrow = c(2, 1))
> title <- "Subject-level frailties, by cluster"
> p <- boxplot(vigndata.fit$frailty$subj1 ~ groups, main = paste(title,
+     "(pain)"), cex.main = 2)
```

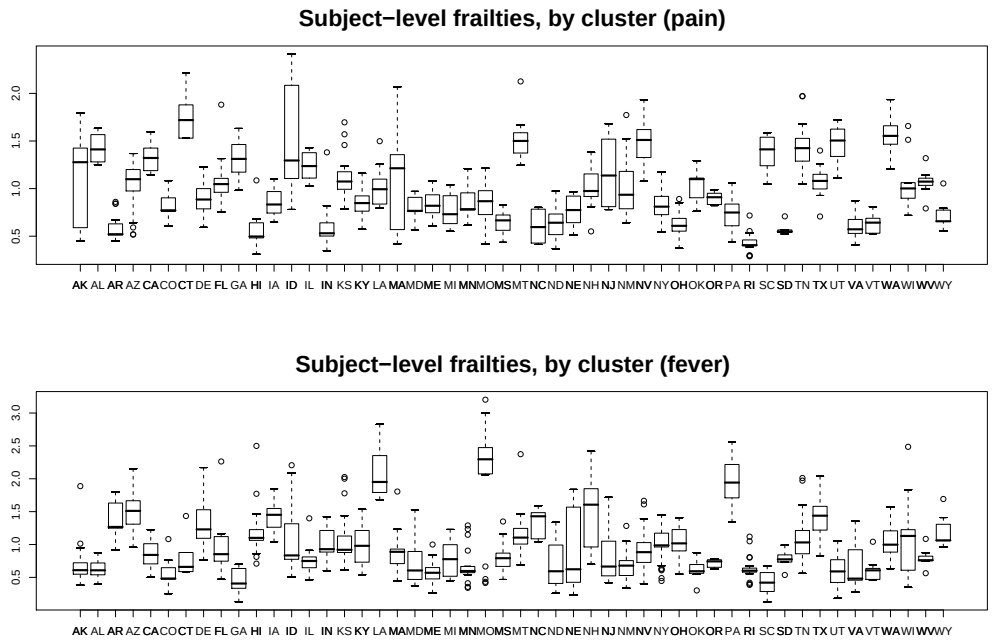

Figure 3: Boxplot of subject-level frailties in each cluster. The cluster-level frailty is the mean of the subject-level frailties.

```
> mtext(side = 1, p$names, at = 1:50, line = 1, cex = 1)
> p <- boxplot(vigndata.fit$frailty$subj2 ~ groups, main = paste(title,
+     "(fever)"), cex.main = 2)
> mtext(side = 1, p$names, at = 1:50, line = 1, cex = 1)
```

Figure 4 shows the estimated subject frailties for pain and fever plotted against each other, to demonstrate the correlation.

```
> plot(vigndata.fit$frailty$subj1, vigndata.fit$frailty$subj2, xlab = "pain",
+     ylab = "fever", main = "Subject-level frailties", pch = 19, asp = 1,
+     xlim = c(0, 3), ylim = c(0, 3))
```

This concludes the demonstration of the `blupsurv` package for analysis of bivariate recurrent event processes. Use of the `unirec` method for analysis of univariate recurrent event processes is analogous.

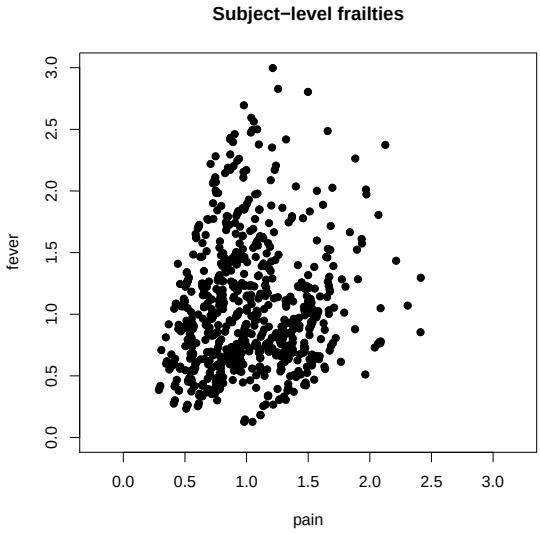

Figure 4: Subject-level frailties for the two processes, showing the correlation.