A quick tutorial of implementing the sensitivity analysis approach for informative visit times in Yiu and Su (2023)

Li Su, Sean Yiu

We provided a quick tutorial to demonstrate how to implement the sensitivity analysis approach for informative visit times in marginal regression analysis proposed in Yiu and Su (2023). Functions to be called in this tutorial were saved in *Functions.R*.

```
source('Functions.R') ## functions to be called
```

1. Data

We simulated longitudinal continuous data based on the data generating mechanism described in Section 3.1 of Yiu and Su (2023). Let $t = 0.01, 0.02, \ldots, 0.49, 5$ be the possible visit times. At each visit time t, two time-varying covariates $X_1(t)$ and $X_2(t)$ from independent normal distributions with mean $-Z$ and unit variance were generated. The group variable Z was generated from a Bernoulli(0.5) distribution at baseline. The outcome $Y(t)$ at t was generated from a Normal distribution with the mean

$$\mathrm{E}\{Y(t) \mid X(t), Z\} = 5 + X_1(t) + X_2(t) - 0.5X_1(t)X_2(t) - 2Z - 0.5t, \tag{1}$$

and a standard deviation of 0.5. For the visit process, We used the Bernoulli distribution to approximate a Cox model as the event/visit rate was set to be low. The visit indicator $dN(t)$ was from a Bernoulli distribution with success probability $\min[1, \exp\{-3.05 - 2t + 0.5X_1(t) + 0.5X_2(t) + 0.5X_1(t)X_2(t) + Z + 0.3Y(t)\}]$. Note that the visit process depended on the current outcome $Y(t)$, therefore the visiting at random assumption was violated.

There were 500 patients in the simulated dataset. Below were the first six rows of these data saved in the data.frame *DATA*. 'ID' was subject ID; 't_start' and 't_stop' were the start and end of the risk interval for the visit process. 'status' was the visit indicator. 'Z' was the baseline group indicator. 'Y' was the longitudinal outcome. 'X1' and 'X2' were time-varying covariates and 'X1X2' were their interaction. The observed data only contained 9694 records from those who made a visit (i.e. with 'status=1').

```
##   ID t_start t_stop status         Y Z           X1         X2         X1X2
##    1    0.00   0.01      0 4.2518529 1 -2.220417856  1.3512131 -3.000257758
##    1    0.01   0.02      1 4.4695479 1  0.001862462  1.0340603  0.001925898
##    1    0.02   0.03      0 0.4887839 1 -0.876658328 -0.7298765  0.639852334
##    1    0.03   0.04      0 2.8243704 1 -1.152686696  0.4773070 -0.550185420
##    1    0.04   0.05      0 0.4346812 1 -1.698453279 -0.7681942  1.304741904
##    1    0.05   0.06      0 1.4673657 1 -1.068713163 -0.2649458  0.283151017
```

2. Marginal model

We were interested in estimating the regression coefficients β_1 and β_2 in the model for the marginal mean of the outcome $\mathrm{E}\{Y(t) \mid Z, t\} = \beta_0 + \beta_1 Z + \beta_2 t$. The true values of β_1 and β_2 were -4.5 and -0.5, respectively, which were obtained by averaging out $X_1(t)$ and $X_2(t)$ from the model in (1).

3. Estimators of β_1 and β_2

For all estimators of β_1 and β_2 except the naive estimator, we assumed that the selection function $\phi Y(t)$ was correctly specified. In our case $\phi = 0.3$. In practice, we can set ϕ at plausible values to assess the sensitivity of substantive conclusions to violations of the visiting at random assumption.

3.1 The naive estimator without inverse intensity weighting Without weighting, we can fit a linear model to the observed data.

```
vis_ind<-which(DATA$status==1)
ugeemod<-lm(Y~Z+t_stop,data=DATA[vis_ind,])
ugeeest<-ugeemod$coef
print(summary(ugeemod))
```

```
##
## Call:
## lm(formula = Y ~ Z + t_stop, data = DATA[vis_ind, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9707 -0.4732  0.0426  0.5416  3.8268
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.28849    0.01403  448.21   <2e-16 ***
## Z           -2.96147    0.02730 -108.49   <2e-16 ***
## t_stop      -0.16210    0.01349  -12.01   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8814 on 9691 degrees of freedom
## Multiple R-squared:  0.5484, Adjusted R-squared:  0.5483
## F-statistic:  5885 on 2 and 9691 DF,  p-value: < 2.2e-16
```

The naive estimates of β_1 and β_2 were -2.96 and -0.16, respectively. The naive estimator overestimated both the group effect β_1 and the time effect β_2.

3.2 The standard inverse intensity weighted estimator (IIWE) with weights estimated using a Cox model If the selection function was omitted, we can fit a Cox model using the observed covariates. Then we estimated the visit intensities and saved them in *hazest*.

```
  library(survival)
  coxmod<-coxph(formula=Surv(t_start,t_stop,status)~Z+X1+X2+X1X2,data=DATA)
  var_vec<-c("Z","X1","X2","X1X2")
  hazest<-exp(colSums(coxmod$coef*t(as.matrix(DATA[,var_vec]))))
  print(coxmod)
```

```
## Call:
## coxph(formula = Surv(t_start, t_stop, status) ~ Z + X1 + X2 +
##     X1X2, data = DATA)
##
##           coef exp(coef)  se(coef)       z        p
## Z     0.143707  1.154546  0.033936   4.235 2.29e-05
## X1    1.312661  3.716050  0.013484  97.352  < 2e-16
## X2    1.310143  3.706703  0.013256  98.831  < 2e-16
## X1X2 -0.134631  0.874038  0.009466 -14.222  < 2e-16
```

```
##
## Likelihood ratio test=31628  on 4 df, p=< 2.2e-16
## n= 250000, number of events= 9694
```

We used the inverse of the estimated visit intensity as weights in the linear model.

```
  wgeemod_noselect<-lm(Y~Z+t_stop,weights=1/hazest[vis_ind],data=DATA[vis_ind,])
  wgeeest_noselect<-wgeemod_noselect$coef
  print(summary(wgeemod_noselect))
```

```
##
## Call:
## lm(formula = Y ~ Z + t_stop, data = DATA[vis_ind, ], weights = 1/hazest[vis_ind])
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -70.503   0.325   0.474   0.698   9.458
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.05069    0.03988  126.66   <2e-16 ***
## Z           -5.42840    0.04311 -125.91   <2e-16 ***
## t_stop      -0.60022    0.04122  -14.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.61 on 9691 degrees of freedom
## Multiple R-squared:  0.6274, Adjusted R-squared:  0.6273
## F-statistic:  8159 on 2 and 9691 DF,  p-value: < 2.2e-16
```

The standard IIWE estimates of β_1, β_2 were -5.43 and -0.6, respectively. The estimate of β_1 was close to the true value, but β_2 was underestimated.

We then fit a Cox model with the observed covariates and the correct selection function. Following the sensitivity analysis approach described in Sections 2.2 and 2.4 in Yiu and Su (2023), we set the inverse of the selection function, $\exp\{-0.3Y(t)\}$, as the case weights w when the visit indicator *status=1*. Note that *w=1* if *status=0*. An offset term $-\log[\exp\{-0.3Y(t)\}]$ was created to prevent the *coxph* function from recalculating the weighted sums of the covariates in the score functions of the Cox model using the case weights w. We therefore were able to use the weighted score functions in equation (2.8) of Yiu and Su (2023) to estimate the rest of the parameters in the Cox model.

```
DATA$w <- exp(-0.3*DATA$Y*DATA$status)
DATA$of <- -log(DATA$w)
coxmod <- coxph(formula=Surv(t_start,t_stop,status)~Z+X1+X2+X1X2+offset(of),
              data=DATA, weight=w, ties='breslow')
print(summary(coxmod))
```

```
## Call:
## coxph(formula = Surv(t_start, t_stop, status) ~ Z + X1 + X2 +
##     X1X2 + offset(of), data = DATA, weights = w, ties = "breslow")
##
##   n= 250000, number of events= 9694
##
##           coef exp(coef)  se(coef) robust se      z Pr(>|z|)
## Z    0.7836767 2.1895076 0.0625412 0.0314786 24.896   <2e-16 ***
## X1   0.8989778 2.4570901 0.0264435 0.0195792 45.915   <2e-16 ***
## X2   0.8880514 2.4303892 0.0262115 0.0215577 41.194   <2e-16 ***
```

```
## X1X2 0.0009129 1.0009133 0.0202699 0.0195438  0.047    0.963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      exp(coef) exp(-coef) lower .95 upper .95
## Z        2.190     0.4567    2.0585     2.329
## X1       2.457     0.4070    2.3646     2.553
## X2       2.430     0.4115    2.3298     2.535
## X1X2     1.001     0.9991    0.9633     1.040
##
## Concordance= 0.957  (se = 0.004 )
## Likelihood ratio test= 3360  on 4 df,   p=<2e-16
## Wald test            = 6471  on 4 df,   p=<2e-16
## Score (logrank) test = 5028  on 4 df,   p=<2e-16,   Robust = 7646  p=<2e-16
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

Then we saved the exponential of the linear predictor function of the fitted Cox model in *hazest*. Together with the specified selection function, we estimated the inverse visit intensities and saved in *iiweight*, which were then included in the linear model for the observed longitudinal continuous data as weights.

```r
var_vec<-c("Z","X1","X2","X1X2" )
hazest<-exp(colSums(coxmod$coef*t(as.matrix(DATA[,var_vec]))))
iiweight<-1/hazest[vis_ind]*exp(-0.3*DATA$Y[vis_ind])
wgeemod<-lm(Y~Z+t_stop,weights=iiweight,data=DATA[vis_ind,])
wgeeest<-wgeemod$coef
print(summary(wgeemod))
```

```
##
## Call:
## lm(formula = Y ~ Z + t_stop, data = DATA[vis_ind, ], weights = iiweight)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -28.5239   0.1538   0.2197   0.3052   2.9138
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.17818    0.03655   141.7   <2e-16 ***
## Z           -5.33809    0.04047  -131.9   <2e-16 ***
## t_stop      -0.42120    0.03794   -11.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6951 on 9691 degrees of freedom
## Multiple R-squared:  0.6444, Adjusted R-squared:  0.6443
## F-statistic:  8781 on 2 and 9691 DF,  p-value: < 2.2e-16
```

The standard IIWE estimates of β_1, β_2 now became -5.34 and -0.42, respectively.

3.3 The IIWE with the balancing weights To obtain the balancing weights proposed in Yiu and Su (2023), we first estimated the increments of cumulative hazard function using the Breslow estimator, where the visit indicator *status* was multiplied by $\exp\{-0.3Y(t)\}$ to account for the impact of the selection function. These estimates were saved in *haz_cont*.

```r
DATA$status2<-DATA$status*exp(-0.3*DATA$Y)

DATA_list_status<-split(DATA$status2,DATA$ID)
no_of_events<-Reduce(`+`,DATA_list_status)

DATA_list_hazest<-split(hazest,DATA$ID)
sum_haz<-Reduce(`+`,DATA_list_hazest)

haz_cont<-no_of_events/sum_haz  ### Breslow estimates of cumulative hazard
```

The covariates to be balanced in the population who made a visit were saved in the matrix *DesignMat_vis*, which included Z, $X_1(t)$, $X_2(t)$, $X_1(t)X_2(t)$ as well as the time variable t and its interaction with other covariates. The covariate means for the at-risk population were saved in *constrain*. The inverse of the selection function was saved in *offset*. We used the function *bal_fit_fun_sa* to estimate the balancing weights with *DesignMat_vis*, *constrain* and *offset* as inputs. We then applied these weights in the linear model.

```r
DesignMat_int<-as.matrix(DATA[,var_vec])
DesignMat<-cbind(1,DATA$t_start, DesignMat_int, DesignMat_int*DATA$t_start)


offset<-exp(-0.3*DATA$Y[vis_ind])
# covariate means for the at-risk population
constrain<-colSums(DesignMat*rep(haz_cont,no_of_pat))
DesignMat_vis<-DesignMat[vis_ind,]


Bal_weights<-bal_fit_fun_sa(DesignMat_vis,constrain, offset)

bal_geemod<-lm(Y~Z+t_stop,weights=Bal_weights,data=DATA[vis_ind,])
bal_geeest<-bal_geemod$coef
print(summary(bal_geemod))
```

```
##
## Call:
## lm(formula = Y ~ Z + t_stop, data = DATA[vis_ind, ], weights = Bal_weights)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -20.0529   0.1066   0.1899   0.3111   6.4518
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.96125    0.03202  154.94   <2e-16 ***
## Z           -4.46990    0.03650 -122.45   <2e-16 ***
## t_stop      -0.39605    0.02631  -15.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6421 on 9691 degrees of freedom
## Multiple R-squared:  0.611,  Adjusted R-squared:  0.6109
## F-statistic:  7610 on 2 and 9691 DF,  p-value: < 2.2e-16
```

The IIWE estimate of β_1, β_2 using the balancing weights are -4.47 and -0.4, respectively. The estimate of β_1 was closer to the true value than that from the IIWE with the weights estimated by the Cox model. The estimates of β_2 were similar.

4. Confidence intervals

Bootstrap and jackknife confidence intervals for β_1, β_2 can be constructed. In particular, jackknife can be useful when there are convergence issues for estimating the weights due to ill-conditioned matrices in a particular bootstrap sample. Specifically, let n be the total number of patients. We can leave out the ith patient's data in the ith jackknife sample ($i = 1, \ldots, n$). The weight estimation and estimation of parameters (e.g. β_1, β_2) are then repeated for the ith jackknife sample. Let $\hat{\beta}^J_{k,i}$ denote the ith jackknife estimate of β_k ($k = 1, 2$) . We calculate the jackknife standard error of β_k as

$$\frac{1}{n(n-1)} \sum_{i=1}^{n} (\hat{\beta}^J_{k,i} - \bar{\beta}^J_k)^2, \qquad k = 1, 2$$

where $\bar{\beta}^J_k = \sum_{i=1}^{n} \hat{\beta}^J_{k,i}/n$. 95% Wald confidence intervals are then constructed using the jackknife standard errors.