A quick tutorial of implementing the sensitivity analysis approach for informative visit times in Yiu and Su (2024)

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 (2024). 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 of the Supplementary Materials of Yiu and Su (2024). Let $t = 0.01, 0.02, \ldots, 0.49, 5$ be the possible visit times. At each visit time t, two time-varying covariates $Z_1(t)$ and $Z_2(t)$ from independent normal distributions with mean $-X$ and unit variance were generated. The group variable X 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

$$\mathrm{E}\{Y(t) \mid X, Z(t)\} = 5 + Z_1(t) + Z_2(t) - 0.5Z_1(t)Z_2(t) - 2X - 0.5t, \tag{1}$$

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 $\min[1, \exp\{-3.05 - 2t + 0.5Z_1(t) + 0.5Z_2(t) + 0.5Z_1(t)Z_2(t) + X + 0.3Y(t)\}]$. 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 data set. 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. 'X' was the baseline group indicator. 'Y' was the longitudinal outcome. 'Z1' and 'Z2' were time-varying covariates and 'Z1Z2' were their interaction. The observed data only contained 6151 records from those who made a visit (i.e. with 'status'=1).

```
##   ID t_start t_stop status         Y X           Z1         Z2         Z1Z2
##    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      1 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 $\mathrm{E}\{Y(t) \mid X, t\} = \beta_0 + \beta_1 X + \beta_2 t$. The true values of β_1 and β_2 were -4.5 and -0.5, respectively, which were obtained by averaging out $Z_1(t)$ and $Z_2(t)$ from the model in~(**??**).

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; see Section 5 for details of calibrating the range of ϕ against observed information.

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

```
##
## Call:
## lm(formula = Y ~ X + t_stop, data = DATA[vis_ind, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3012 -0.6093  0.1861  0.8318  4.4389
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.17648    0.02840 217.490  < 2e-16 ***
## X           -4.10194    0.04240 -96.751  < 2e-16 ***
## t_stop      -0.27626    0.03617  -7.637 2.56e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.414 on 6148 degrees of freedom
## Multiple R-squared:  0.6037, Adjusted R-squared:  0.6036
## F-statistic:  4683 on 2 and 6148 DF,  p-value: < 2.2e-16
```

The naive estimates of β_1 and β_2 were -4.1 and -0.28, 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 only. Then we estimated the visit intensities and saved them in *hazest*.

```
library(survival)
coxmod<-coxph(formula=Surv(t_start,t_stop,status)~X+Z1+Z2+Z1Z2,ties='breslow',data=DATA)
var_vec<-c("X","Z1","Z2","Z1Z2")
hazest<-exp(colSums(coxmod$coef*t(as.matrix(DATA[,var_vec]))))
print(coxmod)
```

```
## Call:
## coxph(formula = Surv(t_start, t_stop, status) ~ X + Z1 + Z2 +
##     Z1Z2, data = DATA, ties = "breslow")
##
##        coef exp(coef) se(coef)      z        p
## X   0.28163   1.32528  0.03642  7.732 1.06e-14
## Z1  0.68264   1.97909  0.01399 48.793  < 2e-16
## Z2  0.69986   2.01348  0.01372 51.017  < 2e-16
```

```
## Z1Z2 0.11780   1.12502  0.01052 11.196  < 2e-16
##
## Likelihood ratio test=8675  on 4 df, p=< 2.2e-16
## n= 250000, number of events= 6151
```

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

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

```
##
## Call:
## lm(formula = Y ~ X + t_stop, data = DATA[vis_ind, ], weights = 1/hazest[vis_ind])
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -27.9997   0.4071   0.7251   1.1718   6.4337
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.98006    0.04794 103.874  < 2e-16 ***
## X           -4.95653    0.05478 -90.485  < 2e-16 ***
## t_stop      -0.30363    0.05803  -5.232 1.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.006 on 6148 degrees of freedom
## Multiple R-squared:  0.5721, Adjusted R-squared:  0.572
## F-statistic:  4111 on 2 and 6148 DF,  p-value: < 2.2e-16
```

The standard IIWE estimates of β_1, β_2 were -4.96 and -0.3, respectively. The estimate of β_1 was now overestimated, 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 (2024), 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*. If *status = 1*, an offset term $-\log[\exp\{-0.3Y(t)\}]$ was created, while if *status = 0*, the offset term was set at zero. Using offset terms was 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 estimating equations in (9) of Yiu and Su (2024) 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)~X+Z1+Z2+Z1Z2+offset(of),
              data=DATA, weight=w, ties='breslow')
print(summary(coxmod))
```

```
## Call:
## coxph(formula = Surv(t_start, t_stop, status) ~ X + Z1 + Z2 +
##     Z1Z2 + offset(of), data = DATA, weights = w, ties = "breslow")
##
##   n= 250000, number of events= 6151
##
##         coef exp(coef) se(coef) robust se     z Pr(>|z|)
## X    0.86671   2.37907  0.06244   0.03931 22.05   <2e-16 ***
```

```
## Z1   0.36630   1.44239  0.02378   0.02012 18.21   <2e-16 ***
## Z2   0.38836   1.47456  0.02377   0.01872 20.75   <2e-16 ***
## Z1Z2 0.34431   1.41101  0.01346   0.01930 17.84   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##      exp(coef) exp(-coef) lower .95 upper .95
## X        2.379     0.4203     2.203     2.570
## Z1       1.442     0.6933     1.387     1.500
## Z2       1.475     0.6782     1.421     1.530
## Z1Z2     1.411     0.7087     1.359     1.465
##
## Concordance= 0.782  (se = 0.011 )
## Likelihood ratio test= 803.8  on 4 df,   p=<2e-16
## Wald test            = 2319  on 4 df,   p=<2e-16
## Score (logrank) test = 974  on 4 df,   p=<2e-16,   Robust = 2368  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 the inverse intensity weights.

```
var_vec<-c("X","Z1","Z2","Z1Z2" )
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~X+t_stop,weights=iiweight,data=DATA[vis_ind,])
wgeeest<-wgeemod$coef
print(summary(wgeemod))
```

```
##
## Call:
## lm(formula = Y ~ X + t_stop, data = DATA[vis_ind, ], weights = iiweight)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9990   0.1979   0.3398   0.5201   4.0923
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.98841    0.04433 112.530  < 2e-16 ***
## X           -4.65891    0.05149 -90.476  < 2e-16 ***
## t_stop      -0.31722    0.05120  -6.196 6.17e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8896 on 6148 degrees of freedom
## Multiple R-squared:  0.5726, Adjusted R-squared:  0.5725
## F-statistic:  4119 on 2 and 6148 DF,  p-value: < 2.2e-16
```

The standard IIWE estimates of β_1, β_2 now became -4.66 and -0.32, respectively.

3.3 The IIWE with the balancing weights To obtain the balancing weights proposed in Yiu and Su (2024), 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*.

```
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 observed covariates to be balanced in the population who made a visit were saved in the matrix *DesignMat_vis*, which included X, $Z_1(t)$, $Z_2(t)$, $Z_1(t)Z_2(t)$ 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.

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

```
##
## Call:
## lm(formula = Y ~ X + t_stop, data = DATA[vis_ind, ], weights = Bal_weights)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2219   0.1902   0.3285   0.5214   6.2615
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.98342    0.04347 114.648   <2e-16 ***
## X           -4.47495    0.05003 -89.441   <2e-16 ***
## t_stop      -0.47622    0.04779  -9.965   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8802 on 6148 degrees of freedom
## Multiple R-squared:  0.5685, Adjusted R-squared:  0.5683
## F-statistic:  4049 on 2 and 6148 DF,  p-value: < 2.2e-16
```

The IIWE estimate of β_1, β_2 using the balancing weights are -4.47 and -0.48, respectively. The estimates of β_1 and β_2 were now closer to the true values than those from the IIWE with the weights estimated by the Cox model.

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 ($i = 1, \dots, n$). 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

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

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.

5. Calibrating the range of the sensitivity parameter

We have demonstrated how to implement the inverse intensity weighted estimators proposed in Yiu and Su (2024) for a fixed value of the sensitivity parameter. In this section, we demonstrate how to calibrate the range of the sensitivity parameter against the information contained in the observed history.

We first estimate the explained variation by the two time-varying covariates $Z_1(t)$ and $Z_2(t)$, the group variable X and time t in the Cox model assuming visiting at random (VAR). As Cox models with time-varying covariates can be approximated by pooled logistic models (see Section 2.3 of Yiu and Su (2024) for details), we can use the formula (15) in Franks et al. (2020) for the variance explained by covariates X in a logistic model:

$$\rho^2_X = \frac{\mathrm{var}(m(X))}{\mathrm{var}(m(X)) + \pi^2/3},$$

where $m(X)$ is the linear predictor in a logistic model. We apply this formula to the fitted Cox model under the visiting at random assumption in Section 3.2, where the equivilant linear predictor is

$$m(\boldsymbol{Z}, t) = \log\{\hat{\lambda}_0(t)\Delta(t)\} + \hat{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{Z},$$

where $\boldsymbol{Z} = (Z_1(t), Z_2(t), X)$, and $\hat{\lambda}_0(t)$ are the baseline intensity estimates, $\Delta(t)$ is the interval length of the counting process format data `t_start,t_stop)` and $\hat{\boldsymbol{\gamma}}$ are the regression coefficient estimates in the Cox model.

```
exp.var<-predict(coxmod, type='expected')
exp.prob<-exp.var*0.01

varmx<-var(log(exp.prob[exp.var>0]))
rhox2=varmx/(varmx+pi^2/3)
rhox2
```

```
## [1] 0.4937815
```

The variation explained by the covariates and time t in the Cox model under VAR, $\rho^2_{\boldsymbol{Z},t}$, is 0.494.

Now we obtain the variation explained by the null Cox model without any covariates.

```
coxmod.null<-coxph(formula=Surv(t_start,t_stop,status)~1,ties='breslow', data=DATA)
exp.null<-predict(coxmod.null, type='expected')
exp.null.prob<-exp.null*0.01
```

```r
varmx.null<-var(log(exp.null[exp.null>0]))
rhox2.null=varmx.null/(varmx.null+pi^2/3)
rhox2.null
```

```
## [1] 0.4026524
```

The variation explained by the null model including time t only, ρ_t^2 is 0.403.

Next the partial variance explained by $\boldsymbol{Z}$ given t (formula (16) in Franks et al, 2020) is

$$\rho^2_{\boldsymbol{Z}|t} = \frac{\rho^2_{\boldsymbol{Z},t} - \rho_t^2}{1 - \rho_t^2}.$$

```r
rhostar=(rhox2-rhox2.null)/(1-rhox2.null)
rhostar
```

```
## [1] 0.1525562
```

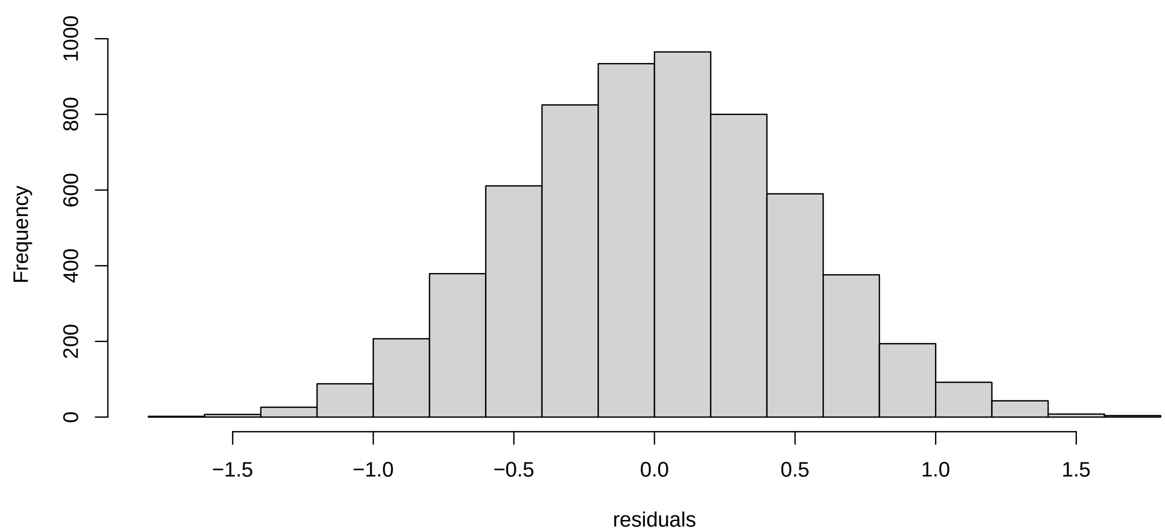
And in this data example, $\rho^2_{\boldsymbol{Z}|t}$ is 0.153.

Suppose that we assume that the additional variation explained by $Y(t)$ given $\boldsymbol{Z}$ is no more than the partial variance explained by $\boldsymbol{Z}$, so that $\rho^2_{Y(t)|\boldsymbol{Z},t} \le \rho^2_{\boldsymbol{Z}|t}$. We can determine the range of the sensitivity parameter ϕ by the formula in (18) of Franks et al. (2020),

$$|\phi| = \frac{1}{\sigma_r}\sqrt{\frac{\rho^2_{Y(t)|\boldsymbol{Z},t}}{1-\rho^2_{Y(t)|\boldsymbol{Z},t}}\{\mathrm{var}(m(\boldsymbol{Z},t)) + \pi^2/3\}},$$

where $\sigma_r = \sqrt{E[var\{Y(t) \mid \boldsymbol{Z}, t\}]}$ (the subscript r stands for *residual*). σ_r is not directly estimable because $Y(t)$ is only observed when $dN(t) = 1$. However, in the setting of our data example, $\sqrt{E[var\{Y(t) \mid \boldsymbol{Z}, t\}]} = \sqrt{var\{Y(t) \mid \boldsymbol{Z}, t, dN(t) = 1\}}$ because $Y(t)$ follows a homoscedastic model and the residual standard deviation is independent of $\boldsymbol{Z}, t, dN(t)$. Therefore, we can use the residual standard deviation in a regression model for observed $Y(t)$ given $\boldsymbol{Z}, t$ to estimate σ_r.

```r
library(splines)
RI_mod<-lm(Y~ns(t_stop, df=5)+X+Z1+Z2+Z1Z2,  data=DATA[which(DATA$status==1),])
hist(RI_mod$residuals, xlab='residuals', main='')
```



```
sdYX<-sd(RI_mod$residuals)
phi=1/sdYX*sqrt(rhostar/(1-rhostar)*(varmx+pi^2/3))
```

The estimated σ_r is 0.498. The corresponding $|\phi|$ for $\rho^2_{Y(t)|\boldsymbol{Z},t}$ set at 0.153 is 2.17, which is larger than 0.3, the true value of ϕ. This is because we assume that the partial variation explained by $Y(t)$ can be as large as the partial variation explained by $\boldsymbol{Z}$, but in fact $\boldsymbol{Z}$ are stronger predictors of the visit intensity than $Y(t)$. Thus in the true data-generating mechanism, the additional variation by $Y(t)$ is less than the variation already explained by $\boldsymbol{Z}$.