UC2 Sentinel Aim2: Cox model and Generalized Raking

Noorie Hyun

2024-`11-15

This markdown file includes example code for fitting a Cox model using the generalized raking approach to address a missing confounding factor. The fitted model is a Cox proportional hazards model:

$$\lambda(t|x_i, ps_i) = \lambda_0(t)\exp(\beta x_i + \sum_{k=1}^{K} \gamma_k B_k(ps_i)$$

, where x_i *is group indicator (1=Covid cohort, 0=Flu cohort).* ps_i is a propensity score with respect to the group factor. $*B_k(\cdot)$ is a basis function for splines.

The propensity score model incorporates BMI values. However, the calculated propensity scores are partially missing because BMI values are observed only for a subset of the cohort. To address this, we apply the generalized raking approach to an inverse probability-weighted Cox model, aiming to reduce standard errors.

Variable Selection for multiple imputations and the missing model (logistic regression) for initial weights in the generalized raking approach

We define "bmi" as median of BMI measured within 90 days prior to hospital admission date

```
### to select variables included in missing model for BMI

in.covar<-c("group","Event","Age","sex","HISPANIC","RACE","zip3","COMORBIDSCO
RE", "NumAV", "NumIP" ,
            setdiff(names(ate.ip)[(grep("covar",names(ate.ip)))],c(paste0("co
var",seq(63,85)),"covarstrat","covarnum","covar87","covar23") ),"covar70","co
var71","COVAR80","COVAR81","COVAR82","COVAR83","fup_dth","bmi","bmi2")

data<-ate.ip[,in.covar]

# ASMD across BMI2 observed and not observed

bmi2.ind<-which(names(data) %in% c("bmi2"))
out<-smdi_asmd(data=data[,-bmi2.ind],covar="bmi" , includeNA=FALSE)
```

```
asmd.out<-data.frame(out$bmi$asmd_table1)
smd.ind<-which(as.numeric(asmd.out$SMD)>=0.1)
smd.var<-row.names(asmd.out)[smd.ind]
smd.var<- word(smd.var, 1)

remove(bmi2.ind, asmd.out, smd.ind, data,in.covar)

# list of variables for Lo re's Propensity Score

ps.list<-c("sex", paste0("covar", seq(1,13)), "covar23", paste0("covar", seq(
28,36)),
           paste0("covar", seq(40,47)),"covar49", "covar51", "covar52", "cova
r58" ,"covar70","covar71", "COMORBIDSCORE", "Age",  "NumAV", "NumIP")



# define list of variables for missing model

bmi.miss.model <-setdiff(unique( union(smd.var, ps.list)),c("covar23"))


data1 <-ate.ip[,c("exposure",ps.list)]
ps.fit<-glm(formula= exposure ~., data=data1,family="binomial")
ps.pred<-predict(ps.fit,type="response")

ate.ip$lo_re_ps<-ps.pred

remove(ps.pred, ps.fit, data1)


## generating basis covariates for original Lo-Re's PS to be included in the
missing model

ps.knots=quantile(ate.ip$lo_re_ps, c(0.5), na.rm=TRUE)
ps.basis<- bSpline(ate.ip$lo_re_ps, knots=ps.knots,   degree = 2)%>%data.fram
e()
nbs<-ncol(ps.basis)

names(ps.basis)<-paste0("ps_bs",seq(1,nbs))
adata <-bind_cols(ate.ip, ps.basis)

## TO screen out variables with P-values>0.1 from the logistic regression wit
h BMI missing indicator outcome

## for bmi
data<-adata[,c("bmi_observed", "followuptime", "fup_event","fup_time_event",p
aste0("ps_bs",seq(1,nbs)),bmi.miss.model )]
```

```
miss.fit0<-glm(formula= bmi_observed ~., data=data,family="binomial")
inda<-which(summary(miss.fit0)$coef[,4]>pvalue.threshold)

exc.var0<-c(rownames(summary(miss.fit0)$coef)[inda],"zip3","group",
            "ps_bs1","ps_bs2","ps_bs3","RACE","HISPANIC","sex")

# BMI initial weight model
bmi.miss.wt.model<-setdiff(bmi.miss.model,exc.var0)


remove(data,adata,exc.var0,miss.fit0,inda)
```

Specify BMI knots for the splines of BMI covariate in the propensity score model

```
#create Basis functions for bmi2
knot<-quantile(ate.ip$bmi2, c(0.25,0.5,0.75), na.rm=TRUE)
print(knot)
```

```
##    25%    50%    75%
## 24.775 29.000 34.725
```

Specify a knot for the splines of propensity score (with BMI) covariate in the outcome model

```
#create Basis functions for bmi2
ps.knots=quantile(ate.ip$lo_re_ps, c(0.5), na.rm=TRUE)
print(ps.knots)
```

```
##       50%
## 0.4892513
```

Data Generation for propensity score with BMI

```
ps.bmi.data<-ate.ip[ate.ip$bmi_observed==1,c("exposure",ps.list,"bmi")]


ps.formula= paste0( "exposure ~",paste(paste(ps.list, "+ "), collapse = '') ,

                    paste0("bSpline(bmi,knot=c(",
                           paste0(knot,collapse=","),
                           "), degree=3)"), collapse="")


ps.bmi.fit<-glm(formula= as.formula(ps.formula) , data=ps.bmi.data,family="bi
nomial")
ps.bmi.pred<-predict(ps.bmi.fit,type="response")
```

```
#Add "ps.bmi" variable to "ate.ip" dataset: note that "ps.bmi" is missing on
the subject of patients with missing BMI values

dat<-ate.ip%>%
left_join(data.frame(PATID = ate.ip$PATID[ate.ip$bmi_observed==1],
                     group= ate.ip$group[ate.ip$bmi_observed==1],
                     ps.bmi=ps.bmi.pred), by=c("group","PATID"))
```

A Cox model with generalized raking

Crude ATE incidence proportions by group

```
ate.ip%>%select(group,fup_event)%>%
  tbl_summary(by=group,
              digits = ~ c(0,1))
```

Characteristic	**cov_ip_ate_dxip** N = 449[1]	**flu_ip_ate_dxip** N = 463[1]
[fup_event]Indicates whether follow-up (at-risk time) ends due to occurrence of outcome of interest	64 (14.3%)	60 (13.0%)

[1]n (%)

Among ppts with BMI measured within (-90 days, 0) prior to hospitalization: crude ATE incidence proportions by group (Covid vs. Flu)

```
ate.ip%>%filter(bmi_observed==1)%>%
  select(group,fup_event)%>%
  tbl_summary(by=group,
              digits = ~ c(0,1))
```

Characteristic	**cov_ip_ate_dxip** N = 139[1]	**flu_ip_ate_dxip** N = 220[1]
[fup_event]Indicates whether follow-up (at-risk time) ends due to occurrence of outcome of interest	21 (15.1%)	29 (13.2%)

[1]n (%)

Application of the generalized raking approach

```
options(width = 350)

dat%>%select(group2,bmi_observed)%>%
  tbl_summary(by=group2,
              digits = ~ c(0,1))
```

Characteristic	COVID N = 449[1]	Flu N = 463[1]
bmi_observed	139 (31.0%)	220 (47.5%)

[1]n (%)

```
#parameter setting
    NimpRaking <- 200 # Number of imputed datasets.
    treatment <- "exposure"
    n.cores<- detectCores()


  #### This function calculated a Cox PH regression influence function, to be
 used below.


  inf.fun.cox <- function(coxfit) {
      infl<- resid(coxfit,type="dfbeta")
      return(infl)
  }



dat<-dat%>%arrange(PATID)
ate.ip<-ate.ip%>%arrange(PATID)


ps.basis<- bSpline(dat$ps.bmi, knots=ps.knots,   degree = 2)%>%data.frame()
nbs<-ncol(ps.basis)

names(ps.basis)<-paste0("ps_bs",seq(1,nbs))

bdata <-bind_cols(ate.ip, ps.basis)


# missing model
    miss_formula<-paste0("bmi_observed ~  ",
                   paste(c("exposure","followuptime","fup_event","fup_time_eve
nt",
                           bmi.miss.wt.model),collapse=" + "),
          paste0("+ bSpline(lo_re_ps, knot=c(", ps.knots , "), degree=2)"),coll
apse="")

# for the outcome model
    formula <- paste("Surv(followuptime, fup_event) ~ exposure +",
    paste0("ps_bs", 1:ncol(ps.basis), collapse="+"))

  # Getting initial sampling (inverse probability) weights
  miss_fit <- glm(formula = as.formula(miss_formula), family="binomial" ,data
 = bdata)
  p_obs <- predict(miss_fit, type = "response")
```

```r
  ip_weights <- 1 / p_obs



  # Counting parameters in outcome model
  initfit <- coxph(formula = as.formula(formula), data = bdata)
  # number of coefficients in Cox model
    coefnum <- length(coef(initfit))


  # Here, we set up a dataset for imputation. We need to impute values of the
 covariates subject to missingness (BMI) for ALL individuals (regardless of o
riginally being observed or not). So, we append the original dataset with row
s for each individual with observed BMI with all data from for those individu
als, but leaving the value of BMI missing; these rows are called "miss_phase2
" below.


    data.mi<-bdata[,c("exposure", "followuptime", "fup_event","fup_time_event
", bmi.miss.model,paste0("ps_bs",seq(1,nbs)),"bmi")]
    phase_2_indx <- (bdata$bmi_observed == 1)
    miss_phase2 <- data.mi[bdata$bmi_observed == 1, ]
    miss_cols <- which(colSums(is.na(data.mi)) > 0)
    miss_phase2[, miss_cols] <- NA
    data.mi2 <- rbind(data.mi, miss_phase2)
    fake_phase_2 <- ((1:nrow(data.mi2)) %in% (nrow(data.mi) + 1):nrow(data.mi
2))
    data.mi <- data.mi2

    # Create N imputed Raking datasets where BMI are imputed for all N indivi
duals.
  init <- mice::mice(data.mi, maxit = 0)
  pred.matrix <- init$predictorMatrix
  set.seed(62347)
  data.imputed <-  futuremice( data=data.mi, m = NimpRaking, n.core = 20, see
d = 62347, predictorMatrix = pred.matrix ,maxit = 70)



  # Estimate value of influence functions by fitting a logistic regression mo
del within each imputed dataset, calculating the resulting influence function
s, and then averaging values of influence functions across imputed datasets.
  infMat_all <- array(data = 0, dim = c(nrow(bdata), coefnum, NimpRaking))
    for (iter in 1:NimpRaking) {
      # Limiting the dataset to imputed data, i.e. removing rows where W was
originally observed.
      imp_init <- mice::complete(data.imputed, iter)
        impData_i <- imp_init[1:nrow(bdata), ]
        impData_i[phase_2_indx, ] <- imp_init[fake_phase_2, ]
```

```
      mifit<-coxph(formula=as.formula(formula), x=TRUE, y=TRUE,data=impData_i
)
      infMat_all[, , iter] <- inf.fun.cox(mifit)
      }
  infMat <- rowMeans(infMat_all, dims = 2)

  # Choose raking variables and add relevant raking variables to the original
 data frame. Here, we use generalized raking with all influence functions;
  # rakeformula = ~ inf1 + ... + infk, where k = # coef fit in regression.

    rakeformula <- "~ inf1"
    for (i in 1:coefnum) {
      varname <- paste0("inf", i)
      bdata$inf <- infMat[, i]
      names(bdata)[names(bdata) == "inf"] <- varname
      if (i > 1) {
        rakeformula <- paste0(rakeformula, "+", varname)
      }
    }

  # Adding the original estimates of the sampling weights to the data frame.
  bdata$ipw_wts <- ip_weights

  # Creating a survey object and calibrating the weights
  mydesign <- survey::twophase(
    id = list(~1, ~1), subset = ~I(bdata$bmi_observed == 1),
    prob = list(NULL, ~I(1 / ipw_wts)), data = bdata,
    pps = list(NULL, poisson_sampling(1 /
  bdata$ipw_wts[bdata$bmi_observed==1]))
  )
  infcal <- survey::calibrate(mydesign, formula = as.formula(rakeformula), ph
ase = 2, calfun = "raking")

  # Fitting the outcome model: conditional treatment effect of interest.
  rakefit <- survey::svycoxph(formula=as.formula(formula) ,design=infcal)

  summary(rakefit)$coef
```

```
## Two-phase sparse-matrix design:
##  survey::calibrate(mydesign, formula = as.formula(rakeformula),
##     phase = 2, calfun = "raking")
## Phase 1:
## Independent Sampling design (with replacement)
## svydesign(ids = ~1)
## Phase 2:
## Sparse-matrix design object:
##  calibrate.pps(phase2, formula, population, calfun = calfun, ...)

##                coef exp(coef)  se(coef) robust se          z  Pr(>|z|)
## exposure  0.2503337 1.2844540 0.3338141 0.2045638  1.2237439 0.2210488
```

```
## ps_bs1    0.7437473 2.1038044 1.0419351 1.0342775  0.7190984 0.4720803
## ps_bs2    0.2100308 1.2337161 0.7475629 0.6271225  0.3349120 0.7376915
## ps_bs3   -1.1558441 0.3147917 1.4331297 1.5973348 -0.7236079 0.4693065
```

```
tbl_regression(rakefit, exponentiate = TRUE)
```

```
## Two-phase sparse-matrix design:
##  survey::calibrate(mydesign, formula = as.formula(rakeformula),
##     phase = 2, calfun = "raking")
## Phase 1:
## Independent Sampling design (with replacement)
## svydesign(ids = ~1)
## Phase 2:
## Sparse-matrix design object:
##  calibrate.pps(phase2, formula, population, calfun = calfun, ...)
```

Characteristic	HR[1]	95% CI[1]	p-value
exposure	1.28	0.86, 1.92	0.2
ps_bs1	2.10	0.28, 16.0	0.5
ps_bs2	1.23	0.36, 4.22	0.7
ps_bs3	0.31	0.01, 7.21	0.5

[1]HR = Hazard Ratio, CI = Confidence Interval

The distributions of initial weights among patinets with missing BMI values

```
summary(ip_weights[bdata$bmi_observed==0])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.224   2.230   3.005   3.142   4.006   6.212
```

```
quantile(ip_weights[bdata$bmi_observed==0], c(0.25, 0.75, seq(0.9,1,by=0.01))
)
```

```
##      25%      75%      90%      91%      92%      93%      94%      95%
   96%      97%      98%      99%     100%
## 2.229528 4.005887 4.596702 4.645437 4.725924 4.811299 4.839857 4.890906 4.
965588 5.148352 5.347886 5.608591 6.212480
```

The distributions of calibrated weights

```
 summary(rakefit$model$`(weights)`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3276  0.7452  0.9500  1.0000  1.1976  2.1766
```

```
quantile(rakefit$model$`(weights)`, c(0.25, 0.75, seq(0.9,1,by=0.01)))
```

```
##       25%       75%       90%       91%       92%       93%       94%
  95%       96%       97%       98%       99%      100%
## 0.7451996 1.1975559 1.5117553 1.5249919 1.5536247 1.5960228 1.6434574 1.67
14431 1.7050890 1.7570423 1.8136057 1.9152098 2.1765911
```