Rihui_BayesOrdinalLogit

Rihui Ou

1/16/2020

Case Study 1: National Collaborative Perinatal Project

Background

The data are taken from the National Collaborative Perinatal Project (CPP). Women were enrolled during pregnancy through different medical centers and then the kids were followed in order to collect both pregnancy and childhood development outcomes. We consider a subsample of 2380 women and children for this analysis, which was studied by [Longnecker et al., 2001]. A particular focus of the Longnecker et al substudy was in assaying serum samples from the original larger study to obtain information on exposures in order to assess the relationship between these exposures to the women and adverse pregnancy and developmental outcomes in their children. Two exposures of particular interest are Dichlorodiphenyldichloroethylene (DDE) and Polychlorinated Biphenyls (PCBs), which are breakdown products in the body of chemicals that have been historically used to treat crops to protect them from predation. These chemicals persist in the environment and are lipophilic, building up in fatty deposits in human tissues. Hence, each of us carries around our own body burden of these chemicals, potentially impacting our health.

The data

The dataset contains demographic variables, such as race, age, and socio-economic index, along with smoking status and concentration doses for DDE and PCBs. In addition, data are available on levels of cholesterol and triglycerides in serum; these variables are relevant since DDE/PCBs are stored in fat and cholesterol/triglycerides provide measurements of the levels of circulating fats (being somewhat informal) in serum.

Goal

The overarching goal of the analysis is to assess how DDE and PCBs relate to risk of premature delivery. Premature delivery is typically defined as a gestational age at delivery of 37 weeks or less, but it is important to note that deliveries occurring right at the cutoff have similar clinical outcomes to full term deliveries, while deliveries occurring substantially less than 37 weeks (early preterm) are associated with substantial risk of short and long term morbidity and mortality. Ideally we would like to infer a causal effect of these exposures on risk of premature deliveries of different severities, while investigating the dose response relationship. However, these data are not collected in a randomized trial but are the result of an observational epidemiology study. Hence, epidemiologists typically focus on assessing associations, while adjusting for covariates that may confound exposure-outcome relationships. In addressing the above interests, it is important to take into account heterogeneity across study centers.

Variable key

gestational_age = gestational age (in weeks)

dde = concentration of dde (ug/dL)

pcb_* = concentration of pcb_* (ng/dL)

albumin = concentration of albumin (g/dL)

cholesterol = concentration of cholesterol (g/dL)

triglycerides = concentration of triglycerides (g/dL) race

score_education

score_income

score_occupation

maternal_age = age of mother

smoking_status = mother smoking

center

The gestational ages are categorized into three categories- gestational ages of (0,34) [35,38) [38,) is catagorized as 0, 1 and 2.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data<-readRDS("~/Downloads/Case-Study-1-master/Longnecker.rds")
data = data %>%
  filter(gestational_age <= 46) %>%
  mutate_at(vars(smoking_status, center),factor) %>%
  dplyr::select(-albumin) %>% #Too many NAs
  mutate(premature=cut(gestational_age, breaks=c(-Inf, 35, 38, Inf), right = FALSE,labels = c(0,1,2)))
#Categorized variable
  #mutate(premature = (gestational_age < 37))
```

Preprocess Data as Alessandro did-summing pcbs

```
# 1) Summing
pcb_sum = apply(as.matrix(data %>% dplyr::select(pcb_028, pcb_052, pcb_074, pcb_105, pcb_118, pcb_153,
# 2) Stanardize and average
my_standardize <- function(x) (x - mean(x, na.rm = T)) / sd(x, na.rm = T)
data = data %>%
  mutate_at(vars(starts_with("pcb")), my_standardize) %>% # standardize pcb's to give them all equal we
  rowwise() %>%
  mutate(pcb_mean = mean(c(pcb_028, pcb_052, pcb_074, pcb_105, pcb_118, pcb_153, pcb_170, pcb_138, pcb_
  ungroup
data$pcb_sum = pcb_sum
```

Calculate the lipid index.

```
data$lipids = 2.27*data$cholesterol +  data$triglycerides + 0.623 #Compute Lipisds#
data$dde_lipid = data$dde/data$lipids
data$pcb_lipid = data$pcb_sum/data$lipids
```

The cumulative logistics model is fitted.

```
data=na.omit(data)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
cumlogit_model_lipid=polr(premature ~ dde_lipid + pcb_lipid + race + maternal_age + score_occupation +
     contrasts = NULL, Hess = FALSE, model = TRUE,
     method = c("logistic"))
summary(cumlogit_model_lipid)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = premature ~ dde_lipid + pcb_lipid + race + maternal_age +
##     score_occupation + center + score_income + score_education,
##     data = data, contrasts = NULL, Hess = FALSE, model = TRUE,
##     method = c("logistic"))
##
## Coefficients:
##                        Value Std. Error    t value
## dde_lipid        -3.274e+00   1.964643   -1.66657
## pcb_lipid        -3.423e+01   0.060457 -566.11955
## raceblack        -5.367e-01   0.208153   -2.57819
## raceother        -6.832e-01   0.435957   -1.56722
## maternal_age     -1.757e-04   0.009428   -0.01863
## score_occupation  2.123e-03   0.002394    0.88663
## center10          4.377e-01   0.339635    1.28861
## center15         -1.922e-01   0.310021   -0.61996
## center31          6.412e-01   0.424124    1.51183
## center37         -5.244e-01   0.261127   -2.00822
## center45          3.015e-01   0.308345    0.97770
## center50         -2.233e-01   0.271660   -0.82200
## center55          7.527e-01   0.530103    1.41995
## center60         -7.096e-02   0.303709   -0.23364
## center66          2.938e-01   0.268823    1.09289
## center71          2.126e-01   0.302395    0.70301
## center82         -1.918e-01   0.301022   -0.63720
## score_income      2.345e-03   0.002353    0.99634
## score_education   2.877e-03   0.002637    1.09104
##
## Intercepts:
##     Value     Std. Error t value
## 0|1   -2.9323    0.3642    -8.0520
## 1|2   -1.5374    0.3561    -4.3173
##
## Residual Deviance: 2329.341
## AIC: 2371.341
```

Cumulative Logit model w/o considering lipid is fitted.

```
###Model 1_
library(MASS)
cumlogit_model=polr(premature ~ dde + pcb_sum + race + maternal_age + score_occupation + center + score_
     contrasts = NULL, Hess = FALSE, model = TRUE,
     method = c("logistic"))
summary(cumlogit_model)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = premature ~ dde + pcb_sum + race + maternal_age +
##     score_occupation + center + score_income + score_education,
##     data = data, contrasts = NULL, Hess = FALSE, model = TRUE,
##     method = c("logistic"))
##
## Coefficients:
##                       Value Std. Error t value
## dde               -0.005898   0.002961 -1.9917
## pcb_sum           -0.084292   0.034730 -2.4271
## raceblack         -0.534149   0.207092 -2.5793
## raceother         -0.740156   0.438503 -1.6879
## maternal_age       0.002150   0.009563  0.2248
## score_occupation   0.002393   0.002404  0.9954
## center10           0.462430   0.341339  1.3548
## center15          -0.274178   0.319331 -0.8586
## center31           0.674923   0.427429  1.5790
## center37          -0.556103   0.261478 -2.1268
## center45           0.304374   0.309881  0.9822
## center50          -0.303008   0.276601 -1.0955
## center55           0.719256   0.534590  1.3454
## center60          -0.148825   0.310804 -0.4788
## center66           0.269236   0.269761  0.9981
## center71           0.176025   0.303451  0.5801
## center82          -0.273147   0.309195 -0.8834
## score_income       0.002543   0.002366  1.0749
## score_education    0.002642   0.002647  0.9980
##
## Intercepts:
##     Value   Std. Error t value
## 0|1 -3.0644  0.3694    -8.2946
## 1|2 -1.6645  0.3609    -4.6116
##
## Residual Deviance: 2321.539
## AIC: 2363.539
```

Variable Selection: Using stepwise regression to do variable selection. Our finalized model is "premature ~ dde + pcb_sum + race + score_occupation + center"

```
finalmodel=stepAIC(cumlogit_model)
```

```
## Start:  AIC=2363.54
## premature ~ dde + pcb_sum + race + maternal_age + score_occupation +
##     center + score_income + score_education
##
##                    Df    AIC
## - maternal_age      1 2361.6
## - score_occupation  1 2362.5
## - score_education   1 2362.5
## - score_income       1 2362.7
## <none>                 2363.5
## - dde                1 2365.4
## - race               2 2367.1
## - pcb_sum            1 2367.2
## - center            11 2373.3
##
## Step:  AIC=2361.59
## premature ~ dde + pcb_sum + race + score_occupation + center +
##     score_income + score_education
##
##                     Df    AIC
## - score_education    1 2360.5
## - score_occupation   1 2360.6
## - score_income       1 2360.8
## <none>                 2361.6
## - dde                1 2363.5
## - race               2 2365.1
## - pcb_sum            1 2365.2
## - center            11 2371.4
##
## Step:  AIC=2360.53
## premature ~ dde + pcb_sum + race + score_occupation + center +
##     score_income
##
##                     Df    AIC
## - score_income       1 2360.2
## - score_occupation   1 2360.4
## <none>                 2360.5
## - dde                1 2362.7
## - race               2 2363.7
## - pcb_sum            1 2364.2
## - center            11 2370.9
##
## Step:  AIC=2360.2
## premature ~ dde + pcb_sum + race + score_occupation + center
##
##                     Df    AIC
## <none>                 2360.2
## - score_occupation   1 2361.7
## - dde                1 2362.2
## - race               2 2363.3
## - pcb_sum            1 2363.8
## - center            11 2370.5
```

Model Checking: Use our original dataset as the test dataset. Our model predicts every data point as label "2", that is, no preterm. This is bad because it means we are having the imbalanced label issue. Our model does a bad job in identifying label 1 or label 0.

```r
sum(predict(finalmodel)==2)/length(data$premature)
```

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

###Solving the imbalanced problem I try to subsample the original dataset (I delete 1000 datapoints with label 2 to make the dataset more balanced), this is to say, subsample the majority group in this case to make our dataset more balanced.

```
newdata=data[-sample(which(data$premature==2),800,replace = FALSE),]
newcumlogit_model=polr(premature ~ dde+ pcb_sum + race + maternal_age + score_occupation + center + sco
     method = c("logistic"))
summary(newcumlogit_model)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = premature ~ dde + pcb_sum + race + maternal_age +
##     score_occupation + center + score_income + score_education,
##     data = newdata, method = c("logistic"))
##
## Coefficients:
##                       Value Std. Error t value
## dde               -0.006789   0.003394 -2.0003
## pcb_sum           -0.065945   0.037785 -1.7453
## raceblack         -0.416196   0.230219 -1.8078
## raceother         -0.761000   0.512802 -1.4840
## maternal_age      -0.006976   0.010662 -0.6543
## score_occupation   0.004633   0.002647  1.7504
## center10           0.458694   0.362486  1.2654
## center15          -0.301740   0.354218 -0.8519
## center31           0.695579   0.465562  1.4941
## center37          -0.466594   0.291341 -1.6015
## center45           0.413003   0.340689  1.2123
## center50          -0.224318   0.302240 -0.7422
## center55           0.998783   0.614685  1.6249
## center60          -0.126794   0.336692 -0.3766
## center66           0.171710   0.301734  0.5691
## center71          -0.044337   0.340430 -0.1302
## center82          -0.397692   0.345187 -1.1521
## score_income       0.003743   0.002643  1.4160
## score_education    0.001767   0.002932  0.6026
##
## Intercepts:
##     Value   Std. Error t value
## 0|1 -2.4939  0.4091    -6.0958
## 1|2 -0.8707  0.4005    -2.1741
##
## Residual Deviance: 1794.253
## AIC: 1836.253
```

Things are getting better! We now have a more balanced result.

```
sum(predict(newcumlogit_model)==2)/length(newdata$premature)
```

```
## [1] 0.964182
```

Again, the backward elimination is used to do variable selection.

```
newfinalmodel=stepAIC(newcumlogit_model,direction = "backward")
```

```
## Start:  AIC=1836.25
## premature ~ dde + pcb_sum + race + maternal_age + score_occupation +
##     center + score_income + score_education
##
```

```
##                    Df    AIC
## - score_education   1 1834.6
## - maternal_age      1 1834.7
## <none>                1836.2
## - score_income      1 1836.3
## - race              2 1836.4
## - pcb_sum           1 1837.2
## - score_occupation  1 1837.3
## - dde               1 1838.2
## - center           11 1839.0
##
## Step:  AIC=1834.62
## premature ~ dde + pcb_sum + race + maternal_age + score_occupation +
##     center + score_income
##
##                    Df    AIC
## - maternal_age      1 1833.3
## <none>                1834.6
## - race              2 1834.7
## - score_income      1 1835.0
## - pcb_sum           1 1835.5
## - score_occupation  1 1836.8
## - dde               1 1836.8
## - center           11 1837.6
##
## Step:  AIC=1833.28
## premature ~ dde + pcb_sum + race + score_occupation + center +
##     score_income
##
##                    Df    AIC
## <none>                1833.3
## - race              2 1833.3
## - score_income      1 1833.5
## - pcb_sum           1 1834.7
## - dde               1 1835.5
## - score_occupation  1 1835.8
## - center           11 1836.1
```

Check the CIs for predictors

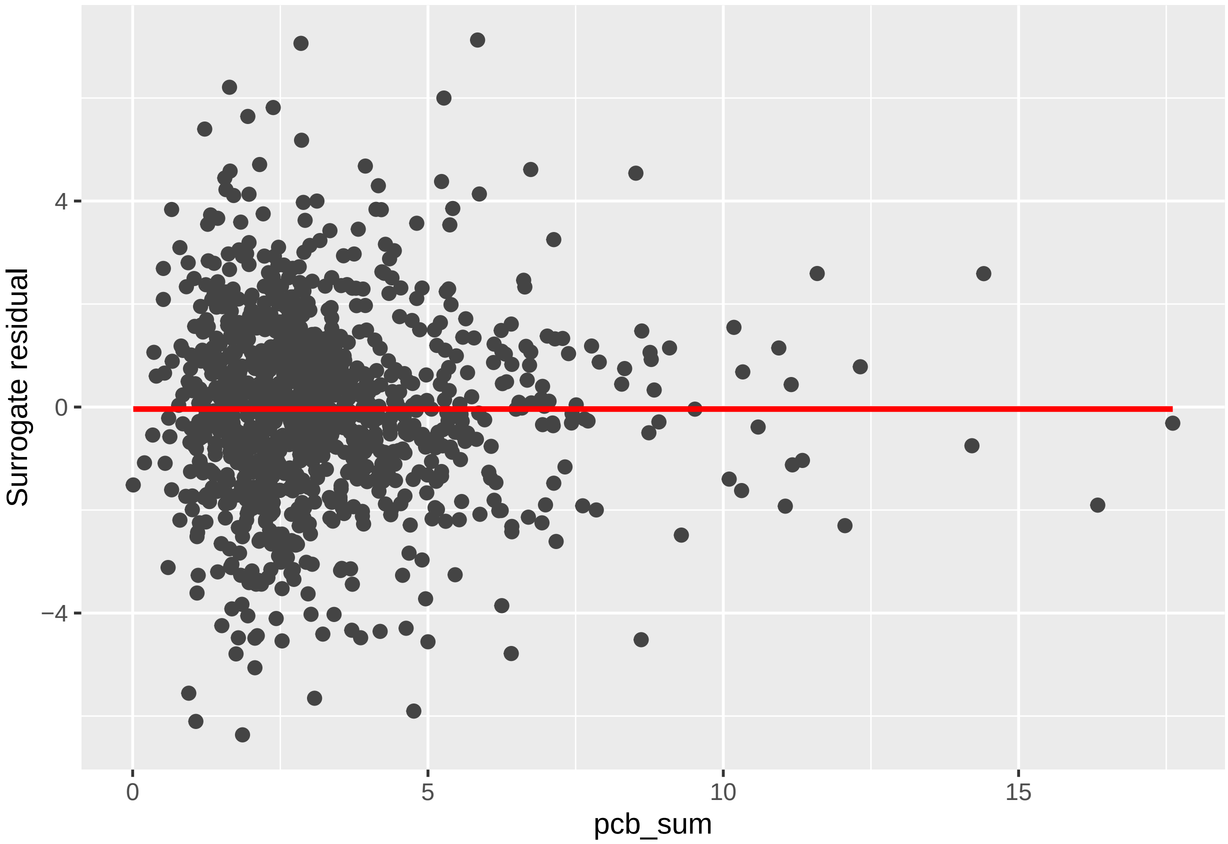
```r
confint(newfinalmodel)
```

```
## Waiting for profiling to be done...

##
## Re-fitting to get Hessian

##                          2.5 %        97.5 %
## dde               -0.0135975993 -0.0003251726
## pcb_sum           -0.1422029235  0.0044736456
## raceblack         -0.8569893314  0.0427779221
## raceother         -1.7501125369  0.2715041969
## score_occupation   0.0003981622  0.0102371473
## center10          -0.1931157173  1.2313164670
## center15          -1.0142419009  0.3676089028
## center31          -0.1998958017  1.6357694297
## center37          -1.0345733007  0.0943650707
## center45          -0.2552297442  1.0660671475
## center50          -0.8120172591  0.3745965188
## center55          -0.1803647018  2.2409958579
## center60          -0.7745671045  0.5474831143
## center66          -0.4061101324  0.7640638916
## center71          -0.7153409114  0.6176950204
## center82          -1.0551459906  0.2874529422
## score_income      -0.0012073403  0.0089749538
```

Check the surrogate residuals vs covariates plot as suggested by https://journal.r-project.org/archive/2018/RJ-2018-004/RJ-2018-004.pdf. The plot looks good.

```r
library(sure)
library(ggplot2)
res=resids(newfinalmodel)
autoplot(res, what = "covariate", x = newdata$pcb_sum, xlab = "pcb_sum")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



The overall predictive accuracy is around 61%.

```r
sum(predict(newfinalmodel)==newdata$premature)/length(newdata$premature)
```

```
## [1] 0.6147144
```

Here's the confusion matrix

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
confusionMatrix(predict(newfinalmodel), newdata$premature, positive = NULL, dnn = c("Prediction", "Label
```

```
## Confusion Matrix and Statistics
##
##           Label
## Prediction   0   1   2
##          0   1   0   1
##          1  14  12  10
##          2 109 264 622
##
## Overall Statistics
##
##                Accuracy : 0.6147
##                  95% CI : (0.5843, 0.6445)
##     No Information Rate : 0.6128
##     P-Value [Acc > NIR] : 0.4628
##
##                   Kappa : 0.0373
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                       Class: 0 Class: 1 Class: 2
## Sensitivity          0.0080645  0.04348   0.9826
## Specificity          0.9988999  0.96830   0.0675
## Pos Pred Value       0.5000000  0.33333   0.6251
## Neg Pred Value       0.8806984  0.73521   0.7105
## Prevalence           0.1200387  0.26718   0.6128
## Detection Rate       0.0009681  0.01162   0.6021
## Detection Prevalence 0.0019361  0.03485   0.9632
## Balanced Accuracy    0.5034822  0.50589   0.5251
```

Check the predictive performance here. Interpretaion will follow after we finalize the model.