Poisson GLM, Cox PH, & degrees of freedom

Michael C. Donohue
Alzheimer's Therapeutic Research Institute
Keck School of Medicine
University of Southern California

December 3, 2015

1 Introduction

We discuss connections between the Cox proportional hazards model and Poisson generalized linear models as described in Whitehead (1980). We fit a sample dataset using coxph() and glm() and show that the model degrees of freedom differ by the number of events.

2 A simple Cox PH example

2.1 Generate data

We generate proportional hazards mixed model data.

```
options(width=75)
library(phmm)

## Loading required package:  survival
## Loading required package:  lattice
## Loading required package:  Matrix

n <- 50      # total sample size
nclust <- 5  # number of clusters
clusters <- rep(1:nclust,each=n/nclust)
beta0 <- c(1,2)
set.seed(13)

Z <-cbind(Z1=sample(0:1,n,replace=TRUE),
          Z2=sample(0:1,n,replace=TRUE),
          Z3=sample(0:1,n,replace=TRUE))
b <- cbind(rep(rnorm(nclust), each=n/nclust),
           rep(rnorm(nclust), each=n/nclust))
Wb <- matrix(0,n,2)
for( j in 1:2) Wb[,j] <- Z[,j]*b[,j]
Wb <- apply(Wb,1,sum)
T <- -log(runif(n,0,1))*exp(-Z[,c('Z1','Z2')]%*%beta0-Wb)
C <- runif(n,0,1)
time <- ifelse(T<C,T,C)
event <- ifelse(T <= C,1,0)
sum(event)
```

```
## [1] 31

phmmd <- data.frame(Z)
phmmd$cluster <- clusters
phmmd$time <- time
phmmd$event <- event
```

2.2 Fit the Cox PH model

```
fit.ph <- coxph(Surv(time, event) ~ Z1 + Z2,
   phmmd, method="breslow", x=TRUE, y=TRUE)

summary(fit.ph)

## Call:
## coxph(formula = Surv(time, event) ~ Z1 + Z2, data = phmmd, x = TRUE,
##     y = TRUE, method = "breslow")
##
##   n= 50, number of events= 31
##
##      coef exp(coef) se(coef)     z Pr(>|z|)
## Z1 0.8549    2.3513   0.3918 2.182  0.02909 *
## Z2 1.0888    2.9708   0.3684 2.955  0.00312 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##    exp(coef) exp(-coef) lower .95 upper .95
## Z1     2.351     0.4253     1.091     5.067
## Z2     2.971     0.3366     1.443     6.116
##
## Concordance= 0.71  (se = 0.055 )
## Rsquare= 0.237   (max possible= 0.984 )
## Likelihood ratio test= 13.55  on 2 df,   p=0.001141
## Wald test            = 13.52  on 2 df,   p=0.001158
## Score (logrank) test = 14.63  on 2 df,   p=0.0006671

fit.ph$loglik[2]

## [1] -95.97131
```

Next we create data to fit an auxilary Poisson model as described in Whitehead (1980) using the pseudoPoisPHMM() function provided in the phmm package. This function also extracts the linear predictors as estimated from the Cox PH model so that we can calculate likelihoods and degrees of freedom.

2.3 Likelihood and degrees of freedom for Poisson GLM from Cox PH parameters

```
ppd <- as.data.frame(as.matrix(pseudoPoisPHMM(fit.ph)))

# pois likelihood
poisl <- c()
eventtimes <- sort(phmmd$time[phmmd$event == 1])
```

```r
for(h in 1:length(eventtimes)){
  js <- ppd$time == eventtimes[h] & ppd$m >= 1  # j star
  j  <- ppd$time == eventtimes[h]
  if(sum(js) > 1) stop("tied event times")
  poisl <- c(poisl,
    ppd[js, "N"]*exp(-1)*exp(ppd[js, "linear.predictors"])/
    sum(ppd[j, "N"]*exp(ppd[j, "linear.predictors"])))
}
```

Poisson likelihood:

```r
sum(log(poisl))

## [1] -66.5633

sum(log(poisl)) - fit.ph$loglik[2]

## [1] 29.40801
```

Poisson degrees of freedom

```r
length(fit.ph$coef) + sum(phmmd$event)

## [1] 33
```

2.4 Fit auxiliary Poisson GLM

We fit an auxiliary Poisson GLM and note that the parameter estimates for z1 and z2 are identical to the coxph() fit, and the likelihood and degrees of freedom are as expected.

```r
ppd$t <- as.factor(ppd$time)
fit.glm <- glm(m~-1+t+z1+z2+offset(log(N)),
  ppd, family=poisson)

summary(fit.glm)

##
## Call:
## glm(formula = m ~ -1 + t + z1 + z2 + offset(log(N)), family = poisson,
##     data = ppd)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9685  -0.7531  -0.5553   0.4293   1.6823
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## t0.000277233256778163  -5.0494     1.0704  -4.717 2.39e-06 ***
## t0.000285092717793308  -5.0035     1.0679  -4.685 2.79e-06 ***
## t0.000382448373472765  -4.9876     1.0683  -4.669 3.03e-06 ***
## t0.00559427171447325   -4.9388     1.0655  -4.635 3.57e-06 ***
## t0.00764335258097282   -4.8875     1.0625  -4.600 4.22e-06 ***
## t0.00808285780728387   -4.8648     1.0635  -4.574 4.78e-06 ***
## t0.0216256697018544    -4.8013     1.0609  -4.526 6.02e-06 ***
```

```
## t0.0219649983261458    -4.7930     1.0622  -4.512 6.41e-06 ***
## t0.0233956453029104    -4.7681     1.0634  -4.484 7.34e-06 ***
## t0.0235837855332384    -4.7069     1.0598  -4.441 8.95e-06 ***
## t0.0237625311885084    -4.6797     1.0612  -4.410 1.03e-05 ***
## t0.027482795605763     -4.6127     1.0572  -4.363 1.28e-05 ***
## t0.0278642961804028    -4.5890     1.0573  -4.340 1.42e-05 ***
## t0.0316525538364514    -4.5401     1.0576  -4.293 1.76e-05 ***
## t0.0357745779481545    -4.5147     1.0578  -4.268 1.97e-05 ***
## t0.0366185731334857    -4.4351     1.0529  -4.212 2.53e-05 ***
## t0.066999301944422     -4.3869     1.0556  -4.156 3.24e-05 ***
## t0.0742904888064418    -4.3572     1.0557  -4.127 3.67e-05 ***
## t0.09491415021304      -4.2493     1.0513  -4.042 5.30e-05 ***
## t0.125132209250348     -4.2151     1.0513  -4.010 6.08e-05 ***
## t0.132722661166308     -4.1798     1.0513  -3.976 7.01e-05 ***
## t0.140357744467437     -4.0667     1.0439  -3.896 9.79e-05 ***
## t0.163527928343998     -3.9258     1.0448  -3.757 0.000172 ***
## t0.193971448733795     -3.7760     1.0443  -3.616 0.000299 ***
## t0.204887967162952     -3.7054     1.0458  -3.543 0.000396 ***
## t0.227852125295401     -3.6459     1.0457  -3.486 0.000490 ***
## t0.266238317485871     -3.5253     1.0513  -3.353 0.000799 ***
## t0.276177426334698     -3.2951     1.0356  -3.182 0.001464 **
## t0.360993505812205     -3.2039     1.0353  -3.095 0.001970 **
## t0.426697507683412     -2.7934     1.0367  -2.694 0.007051 **
## t0.511995413073629     -1.8487     1.0105  -1.830 0.067323 .
## z1                      0.8549     0.3918   2.182 0.029092 *
## z2                      1.0888     0.3684   2.955 0.003123 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1743.184  on 121  degrees of freedom
## Residual deviance:   71.127  on  88  degrees of freedom
## AIC: 199.13
##
## Number of Fisher Scoring iterations: 6
```

```
fit.ph$coef
```

```
##        Z1        Z2
## 0.8549497 1.0888337
```

```
logLik(fit.glm)
```

```
## 'log Lik.' -66.5633 (df=33)
```

```
logLik(fit.glm)[1] - sum(log(poisl))
```

```
## [1] -1.421085e-14
```

The additional parameter estimates correspond to the estimated log baseline hazard, which we verify using the basehaz() function.

```
bh <- basehaz(fit.ph, centered = FALSE)
log(bh$hazard - c(0,bh$hazard[1:(length(bh$hazard)-1)]))[1:10]
```

```
## [1] -5.049378 -5.003546 -4.987633 -4.938810 -4.887479 -4.864823      -Inf
## [8] -4.801254 -4.793001 -4.768072
```

3 Extending to PHMM

3.1 Fit PHMM

```
fit.phmm <- phmm(Surv(time, event) ~ Z1 + Z2 + (Z1 + Z2|cluster),
   phmmd, Gbs = 100, Gbsvar = 1000, VARSTART = 1,
   NINIT = 10, MAXSTEP = 100, CONVERG=90)
summary(fit.phmm)

##
## Proportional Hazards Mixed-Effects Model fit by MCMC-EM
##   Model: Surv(time, event) ~ Z1 + Z2 + (Z1 + Z2 | cluster)
##   Data: phmmd
##   Log-likelihood:
## Conditional     Laplace         RIS
##      -83.32     -122.68     -122.55
##
## Fixed effects: Surv(time, event) ~ Z1 + Z2
##    Estimate Std.Error
## Z1   0.8091    0.5529
## Z2   1.4992    0.5198
##
## Random effects: (Z1 + Z2 | cluster)
## Estimated variance-covariance matrix:
##             (Intercept)     Z1   Z2
## (Intercept)      0.1861 0.0000 0.00
## Z1               0.0000 0.4469 0.00
## Z2               0.0000 0.0000 0.39
##
## Number of Observations: 50
## Number of Groups:  5
```

3.2 Likelihood and degrees of freedom for Poisson GLMM from PHMM parameters

```
ppd <- as.data.frame(as.matrix(pseudoPoisPHMM(fit.phmm)))

poisl <- c()
eventtimes <- sort(phmmd$time[phmmd$event == 1])
for(h in 1:length(eventtimes)){
  js <- ppd$time == eventtimes[h] & ppd$m >= 1  # j star
  j  <- ppd$time == eventtimes[h]
  if(sum(js) > 1) stop("tied event times")
  poisl <- c(poisl,
    ppd[js, "N"]*exp(-1)*exp(ppd[js, "linear.predictors"])/
    sum(ppd[j, "N"]*exp(ppd[j, "linear.predictors"])))
}
```

Poisson likelihood:

```
sum(log(poisl))

## [1] -93.35469

sum(log(poisl)) - fit.phmm$loglik[1]

## Conditional
##   -10.03456
```

Poisson degrees of freedom

```
# Poisson GLMM degrees of freedom  length(unique(x$cluster)) * x$nrandom + x$nfixed
traceHat(fit.phmm, "pseudoPois") # + 2*sum(phmmd$event)

## [1] 6.550302
```

3.3 Fit auxiliary Poisson GLMM

We fit an auxiliary Poisson GLMM, although with a general variance-covariance matrix for the random effects (phmm() only fits models with diagonal variance-covariance matrix).

```
library(lme4)
ppd$t <- as.factor(ppd$time)
fit.lmer <- glmer(m~-1+t+z1+z2+
  (z1+z2|cluster)+offset(log(N)),
  data=ppd, family=poisson)

summary(fit.lmer)$coef

##                          Estimate Std. Error   z value     Pr(>|z|)
## t0.000277233256778163 -5.951196  1.1417661 -5.212272 1.865418e-07
## t0.000285092717793308 -5.806712  1.1269299 -5.152683 2.567864e-07
## t0.000382448373472765 -5.787128  1.1280059 -5.130406 2.891186e-07
## t0.00559427171447325  -5.690766  1.1229759 -5.067576 4.029129e-07
## t0.00764335258097282  -5.583909  1.1181592 -4.993841 5.919028e-07
## t0.00808285780728387  -5.573184  1.1188033 -4.981380 6.313257e-07
## t0.0216256697018544   -5.349824  1.0927283 -4.895841 9.788633e-07
## t0.0219649983261458   -5.346013  1.0932801 -4.889884 1.008954e-06
## t0.0233956453029104   -5.302396  1.0962113 -4.837020 1.318002e-06
## t0.0235837855332384   -5.000402  1.0661190 -4.690285 2.728243e-06
## t0.0237625311885084   -4.936782  1.0685582 -4.620040 3.836662e-06
## t0.027482795605763    -4.905587  1.0685579 -4.590848 4.414492e-06
## t0.0278642961804028   -4.873013  1.0682367 -4.561735 5.073262e-06
## t0.0316525538364514   -4.813772  1.0683795 -4.505676 6.616205e-06
## t0.0357745779481545   -4.763238  1.0690629 -4.455526 8.368770e-06
## t0.0366185731334857   -4.464947  1.0450552 -4.272451 1.933364e-05
## t0.066999301944422    -4.341154  1.0539939 -4.118765 3.809077e-05
## t0.0742904888064418   -4.317542  1.0540464 -4.096160 4.200605e-05
## t0.09491415021304     -4.260188  1.0533735 -4.044328 5.247336e-05
## t0.125132209250348    -4.195529  1.0520827 -3.987832 6.667978e-05
## t0.132722661166308    -4.180512  1.0534484 -3.968407 7.235467e-05
## t0.140357744467437    -4.048693  1.0426772 -3.882978 1.031849e-04
```

```
## t0.163527928343998    -3.850519  1.0370875 -3.712820 2.049626e-04
## t0.193971448733795    -3.567745  1.0305056 -3.462130 5.359175e-04
## t0.204887967162952    -3.447519  1.0308873 -3.344225 8.251286e-04
## t0.227852125295401    -3.389876  1.0315419 -3.286222 1.015409e-03
## t0.266238317485871    -3.257701  1.0390354 -3.135313 1.716708e-03
## t0.276177426334698    -3.081551  1.0296562 -2.992796 2.764347e-03
## t0.360993505812205    -2.861887  1.0292501 -2.780555 5.426604e-03
## t0.426697507683412    -2.398274  1.0201220 -2.350968 1.872466e-02
## t0.511995413073629    -1.640122  1.0049529 -1.632038 1.026714e-01
## z1                     0.752279  0.4528007  1.661391 9.663493e-02
## z2                     1.548775  0.4635595  3.341048 8.346280e-04

fit.phmm$coef

##       Z1       Z2
## 0.809141 1.499185

logLik(fit.lmer)

## 'log Lik.' -100.9787 (df=39)

sum(log(poisl)) - logLik(fit.lmer)[1]

## [1] 7.624055
```

```
log(fit.phmm$lambda)[1:10]

##  [1] -5.903951 -5.782283 -5.759337 -5.660927 -5.551765 -5.542851      -Inf
##  [8] -5.358731 -5.354172 -5.299521
```