HW 3

November 9, 2017

0.0.1 Yue You

Collaborator: Rachael Philips, Thomas Carpenito

1 Problem 1

Show the CAR condition, $x \to Pr(O = o|X = x)$ **for** $x \in \mathcal{C}(o)$ **is constant implies** $Pr(X = x|O = o) = Pr(X = x|x \in \mathcal{C}(o))$**. You may assume all random variables here are discrete for simplicity.**
The CAR condition, $\forall x \in \mathcal{C}(o), Pr(O = o|X = x)$ is constant,

which means the observation $O = o$ is not influenced by values of $X \in \mathcal{C}(o)$ which is taken, only by the fact that X did take a value in $\mathcal{C}(o)$.

want to show: $\forall o,\ x \in \mathcal{C}(o),\ P_r(O = o|\ X = x) = P_r(O = o|\ X \in \mathcal{C}(o))$............................(1)

By conditional independet assumption:

Given $X \in \mathcal{C}(o)$, the events $X = x$ and $O = o$ are conditionally independent.

(1) is equivalent to $\forall x \in \mathcal{C}(o),\ P_r[O = o|\ X = x\ and\ X \in \mathcal{C}(o)] = P_r[O = o|\ X \in \mathcal{C}(o)]$,

By symmetry we have:
$P_r[X = x|\ O = o\ and\ x \in \mathcal{C}(o)] = P_r[O = o|\ X \in \mathcal{C}(o)]$.
Since $P_r[X = x|\ O = o\ and\ x \in \mathcal{C}(o)] = P_r[X = x|\ X \in \mathcal{C}(o)]$
$\therefore P_r(X = x|\ O = o) = P_r[X = x|\ X \in \mathcal{C}(o)]$
$\therefore P_r(O = o|\ X \in \mathcal{C}(o)) = P_r(O = o|\ X = x)$

2 Problem 2

Let $P_{X,\epsilon}$ **be a path through** P_X**, the distribution of the full data,** X**, and having score** $S_1(X)$**. This then defines a path** $P_{P_{X,\epsilon},G}$ **through the observed data distribution,** $P_{P_X G}$**. Show that the scores generated by these paths are** $E[S_1(X)|O = o]$**.** $\frac{d}{d\epsilon}\frac{log(p_{P_{X,\epsilon},G})}{p_{P_X,G}}|_{\epsilon=0}$

$= \frac{d}{d\epsilon}log(p_{P_{X,\epsilon},G})|_{\epsilon=0}$
$= \int \frac{p(O=o|X=x)S_1(x)p(x)dv(x)}{p_{P_X,G}}$
$= \int S_1(x)p(x|O = o)dv(x)$
$= E(S_1(X)|O = o)$

3 Problem 3

Let G_ϵ be a path through G, the distribution of the censoring time, C, given X, having score $S_2(C,X)$. This then defines a path $P_{P_X G_\epsilon}$ through the observed data distribution, $P_{P_X G}$. Show that the scores generated by these paths are $E[S_2(C,X)|O=o]$. $\frac{d}{d\epsilon}\frac{log(p_{P_X,G_\epsilon})}{p_{P_X,G}}|_{\epsilon=0}$

$= \frac{d}{d\epsilon}log(p_{P_X,G_\epsilon})|_{\epsilon=0}$
$= \int \frac{p(O=o|X=x)S_2(c,x)p(x)dv(x)}{p_{P_X,G}}$
$= \int S_2(c,x)p(x|O=o)dv(x)$
$= E(S_2(X)|O=o)$

4 Problem 4

This problem involves simulating data under a general Cox model. Let's make the assumption we have a conditional hazard of death at time, t, given by $\lambda(t|X) = \lambda_0(t)exp(f_\beta(X))$ where X is a set of covariates and f_β is a function indexed by β, say finite dimensional. Assume the baseline hazard is $\lambda_0(t) = exp(rt)$ for positive r. Given X, what is the distribution of death times? Prove your answer. Let t denote death time.

$S(t|\, x) = exp(-\int_0^t d\Lambda(S|X)) =^{\mathcal{D}} U \sim Uniform(0,1)$
Where
$\int_0^t d\Lambda(S|X)$
$= \int_0^t \lambda(s|x)ds$
$= \int_0^t e^{rs}e^{f_\beta(x)}ds$
$= e^{f_\beta(x)}\frac{1}{r}\int_0^t 1de^{rs}$
$= e^{f_\beta(x)}\frac{1}{r}(e^{rt}-1)$
$\therefore S(t|\, x) = exp(-\int_0^t d\Lambda(S|X))$
$= exp(-e^{f_\beta(x)}\frac{1}{r}(e^{rt}-1)) =^{\mathcal{D}} U \sim Uniform(0,1)$
$\Rightarrow -\frac{1}{r}e^{f_\beta(x)}(e^{rt}-1) =^{\mathcal{D}} log(U)$
$\Rightarrow e^{f_\beta(x)}(e^{rt}-1) =^{\mathcal{D}} -rlog(U))$
$\Rightarrow e^{rt}-1 =^{\mathcal{D}} \frac{-rlog(U)}{e^{f_\beta(x)}}$
$\Rightarrow e^{rt} =^{\mathcal{D}} 1-\frac{rlog(U)}{e^{f_\beta(x)}}$
$\Rightarrow rt =^{\mathcal{D}} log\left(1-\frac{rlog(U)}{e^{f_\beta(x)}}\right)$
$\Rightarrow t =^{\mathcal{D}} \frac{1}{r}log\left(1-\frac{rlog(U)}{e^{f_\beta(x)}}\right)$ where $U \sim Uniform(0,1)$

5 Problem 5

Complete the first problem from LabCox in the lab section of the files on bCourses.

```
In [1]: # load packages
        library(survival)
        library(mvtnorm)
        library(survminer)
        setwd("~/Desktop/ph240b-survival_analysis/HW3/Yue")
        rm(list=ls())
```

```
Loading required package: ggplot2
Loading required package: ggpubr
Loading required package: magrittr
```

```
In [2]: # function that takes in number of iterations,
        # and return an average coverage of 95% CI over the truth
        CI_coverage <- function(n) {
          for(i in 1:n) {
            # draw the W from standard normals
            W1 = rnorm(n)
            W2 = rnorm(n)
            A = rbinom(n, 1, plogis(0.1 + W1 * W2))
            # draw T and C --both generated with random uniforms as perviously described but C
            # be generated anyway independent of T given W for identifiability purposes
            C = -log(runif(n))/(0.01 * exp(0.3 * W1))
            T =  -log(runif(n))/(0.02 * exp(2 * W1 ** 2 - A))
            # Create the survival objec
            S = Surv(time = pmin(T, C), event = (C >= T & T <= 100), type = "right")
            data = data.frame(A = A, W1 = W1, W2 = W2)
            coxfit = coxph(S ~ ., data = data)
            # true value
            truth <- exp(-1)
            # 95% CI for A
            lower <- summary(coxfit)$conf.int[, 3][1]
            upper <- summary(coxfit)$conf.int[, 4][1]
            CI_count <- CI_count +  as.numeric(truth >= lower && truth <= upper)
          }
          return(CI_count/ 1000)
        }
```

```
In [3]: CI_count <- 0
        # coverage
        cat("The average coverage of 1000 95% CI's for A is", 100 * CI_coverage(1000), "%.")
```

```
The average coverage of 1000 95% CI's for A is 0 %.
```

6 Bonus

Assume a CAR model for full data consisting of survival time, censoring time, the continuous baseline covariates and randomly assigned treatment indicator. We have observed data $min(T, C), \Delta$ along with the covariates and treatment indicator. Someome receives a data set of 1000 independent subjects drawn from this model from an RCT and runs a Cox Proportional hazards regression with treatment as the only covariate, showing a significantly negative coefficient. Can you convince this person he may be wrong via simulation? Explain how you set up your simulation and turn in your code to show the results.

```
In [4]: CI_coverage_bonus <- function(N, num_W) {
          for(i in 1:N){
```

```r
    # simulation
    W = list()
    for(j in 1:num_W) {
      W[[j]] = rnorm(1000)
    }
    sum_W = 0
    sum_W_half = 0
    for(j in 1:num_W) {
      sum_W = sum_W + W[[j]]
      if (j < floor(num_W/2) + 1) {
        sum_W_half = sum_W_half + W[[j]]
      }
    }
    A = rbinom(1000, 1, 0.5)
    # we make C and T dependent on covariates and we also
    # make C dependent on A,
    # such that drop out is effected by treatment
    C = abs(sum_W - 10 * A)
    T = abs(sum_W_half)
    S = Surv(time = pmin(T, C), event = (C >= T & T <= 2), type = "right")
    # because T is not dependent on A, the true coefficient for A is
    truth = exp(0)
    # they run a Cox Proportional Hazards regression
    # with treatment as the only covariate
    coxfit = coxph(S ~ ., data = data.frame(A))
    # did they cover the truth in their misspecified model?
    lower <- summary(coxfit)$conf.int[, 3]
    upper <- summary(coxfit)$conf.int[, 4]
    CI_count <- CI_count +  as.numeric(truth >= lower && truth <= upper)
  }
  # calculating the coverage
  return(CI_count/ 1000)
}
```

```r
In [5]: CI_count <- 0
        cat("The average coverage rate of 1000 95% CI for A with 2 confounders is",
            100 * CI_coverage_bonus(1000, 2), "%.\n")
        cat("The average coverage rate of 1000 95% CI for A with 5 confounders is",
            100 * CI_coverage_bonus(1000, 5), "%.\n")
        cat("The average coverage rate of 1000 95% CI for A with 7 confounders is",
            100 * CI_coverage_bonus(1000, 7), "%.\n")
```

```
The average coverage rate of 1000 95% CI for A with 2 confounders is 34.2 %.
The average coverage rate of 1000 95% CI for A with 5 confounders is 67.5 %.
The average coverage rate of 1000 95% CI for A with 7 confounders is 70 %.
```

We set up the simulation by letting C equals the absolute value of sum of all confounders minus 10 times A, and letting T equals the absolute value of the half of first half of confounders.

Thus C and T both depend on confounders, and the survival time depend on confounders. When we only account for A, the coverage we expect is low, as shown by the computation above.