PH C240B: Assignment 2

Rachael Phillips

10/20/2017

Problem 1

Consider the following data generating system for (T, C), a random person's survival time and censoring time in days. T follows an exponential distribution with $\lambda = 0.01$ and C follows a weibull distribution with shape and scale parameters, 4 and 100. Simulate 1000 random draws of $n = 1000$ from this distribution and for each draw compute the Kaplan-Meier estimator for survival curve at times 50, 60 and 70, where we consider the study ending at time 100. Form 95% simultaneous confidence bands from each draw and check simultaneous coverage of the true survival. Use the influence curve for the KM estimator to obtain the CI's for each draw. Report the coverage percentage. Note, use foreach for this as shown in lab3sol.Rnw on bcourses to save some time.

```r
S0 = function(t) 1-pexp(t,.01)

sim.cov = function(n) {
  T = rexp(n, .01)
  C = rweibull(n, 4, scale = 100)

  # record the observed right censored data
  Ttilde = pmin(T,C)
  Delta = T <= C & T < 100

  # do a KM fit
  km = survfit(Surv(Ttilde, Delta, type = "right") ~ 1, type = "kaplan-meier",
               conf.int = .95)

  # make step functions
  km_step <- stepfun(km$time, c(1, km$surv))
  km_lower <- stepfun(km$time, c(1, km$lower))
  km_upper <- stepfun(km$time, c(1, km$upper))

  # order the times and corresponding deltas
  T_ord = Ttilde[order(Ttilde)]
  D_ord = Delta[order(Ttilde)]


  ##### CHANGES MADE ######
  # Get all known death times in order and compute the hazard at such times
  # lambda = km$n.event[D_ord]/km$n.risk[D_ord]
  T_death = T_ord[D_ord]
  nrisk = vapply(T_death, FUN = function(x) sum(Ttilde >= x), FUN.VALUE = 1)
  lambda = 1/nrisk
  #########################

  # compute the prob of surviving up to or past each death time
  Pbar = vapply(T_death, FUN = function(time) mean(Ttilde >= time), FUN.VALUE = 1)
```

```r
  IC_time = function(Ttilde, Delta, time, n) {
    sum((-(T_death <= time)*km_step(time)/(1 - lambda))*
          lambda^2*(1 - lambda)^2/((1 - lambda)^2/n + lambda^2*Pbar)*
          ((Ttilde == T_death & Delta == 1)/lambda - (Ttilde > T_death)/(1 - lambda)))
  }

  # compute the IC matrix with each col an IC
  IC = vapply(c(50,60,70),FUN = function(t) {
    unlist(lapply(1:n, FUN = function(i) {
      IC_time(Ttilde[i], Delta[i], t, n)
    }))
  }, FUN.VALUE = rep(1,n))


  # COMPUTE THE CORRELATION and draw the random three-d normals
  Sigma = cor(IC)
  z = rmvnorm(1e6, c(0,0,0), Sigma)

  # compute the max abs val of each of the 3-d normals then choose the
  # 95th quantile of that vector, which is the simultaneous number of SE's
  z_abs = apply(z, 1, FUN = function(row) max(abs(row)))
  SE_num = quantile(z_abs, .95)

  # Note how the CI is wider when demanding simultaneous coverage
  Cov50 = km_lower(50) <= S0(50) & km_upper(50) >= S0(50)
  Cov60 = km_lower(60) <= S0(60) & km_upper(60) >= S0(60)
  Cov70 = km_lower(70) <= S0(70) & km_upper(70) >= S0(70)

  indy_cov = all(Cov50, Cov60, Cov70)

  # simultaneous for 40, 50, 60
  Sim50 = km_step(50) - SE_num*sd(IC[,1])*sqrt(n-1)/n <= S0(50) &
    S0(50) <= km_step(50) + SE_num*sd(IC[,1])*sqrt(n-1)/n
  Sim60 = km_step(60) - SE_num*sd(IC[,2])*sqrt(n-1)/n <= S0(60) &
    S0(60) <= km_step(60) + SE_num*sd(IC[,2])*sqrt(n-1)/n
  Sim70 = km_step(70) - SE_num*sd(IC[,3])*sqrt(n-1)/n <= S0(70) &
    S0(70) <= km_step(70) + SE_num*sd(IC[,3])*sqrt(n-1)/n

  sim_cov = all(Sim50, Sim60, Sim70)

  return(c(indy_cov = indy_cov, sim_cov = sim_cov))
}

sim.cov(1000)
```

```
## indy_cov  sim_cov
##     TRUE     TRUE
```

```r
registerDoParallel(cores = detectCores())
getDoParWorkers()
```

```
## [1] 4
```

```r
B = 1000
n = 1000
```

```
ALL = foreach(i=1:B,.packages=c("mvtnorm"), .errorhandling = "remove") %dopar% {sim.cov(n)}
res = do.call(rbind, ALL)
colMeans(res)
```

```
## indy_cov  sim_cov
##    0.916    0.950
```

Problem 2

Let (W, A, Y) be the observed data with distribution, $P_0 \in M$ nonparametric. Define the following parameter mapping for $P \in M : \Psi(P) = E_P[(1, 1, W)\beta]$ where $\beta = \Psi^1(P) = \text{argmin}_\delta E_P(Y - (1, A, W)\gamma)^2$.

(a) The empirical distribution, $\mathbf{P}_n$, is the NPMLE for the true distribution. We use $\mathbf{P}_n$ as a plug-in estimator, $\Psi(\mathbf{P}_n)$, for the true parameter, $\Psi(P_0)$. This is called the NPMLE for $\Psi(P_0)$. Derive this estimator's influence curve. You may derive the efficient influence curve, $D^*_\Psi(P)$ first and then your answer is $D^*_\Psi(\mathbf{P}_n)$ This is a valid approach but not the only approach.

Well I fucked this up and forgot to consider the observed data:

We have that $\mathbf{P}_n = \text{argmax}_P P_n log\frac{dP}{d\mu}$ and $\mathbf{P}_{n,\epsilon}$ is a path through $\mathbf{P}_n$ with score $D^*(\mathbf{P}_n)$. Then $\epsilon \to P_n log\frac{d\mathbf{P}_{n,\epsilon}}{d\mu}$ is maximized at $\epsilon = 0$. Thus, $O = P_n \frac{d}{d\epsilon} log\frac{d\mathbf{P}_{n,\epsilon}}{d\mu}|_{\epsilon=0} = P_n D^*(\mathbf{P}_n) = 0$. Define $R_2(P_n, P_0) \equiv \Psi(P_n) - \Psi(P_0) - (P - P_0)D^*(P)$. Therefore, $\Psi(\mathbf{P}_n) - \Psi(P_0) = -P_0 D^*(\mathbf{P}_n) + R_2(\mathbf{P}_n, P_0) = (P_n - P_0)D^*(\mathbf{P}_n) + R_2(\mathbf{P}_n, P_0)$. If $R_2(\mathbf{P}_n, P_0) = o_p(n^{-1/2}), P_0(D^*(\mathbf{P}_n), P_0(D^*(\mathbf{P}_n) - D^*(P_0))^2 \to_P 0$, and $Pr(D^*(\mathbf{P}_n) \in \text{Donsker}) \to 1$ as $n \to \infty$, then $\Psi(\mathbf{P}_n) - \Psi(P_0) = P_n D^*(\mathbf{P}_n) + o_P(n^{-1/2})$.

(b) Consider the following data generating process: W_1, W_2 are standard normals, $Pr(A = 1) = expit(1 + 0.4W_1 - 0.4W_2W_1)$, $Y = (W_1 + W_2)^2 + \epsilon$, $\epsilon \sim N[0, 1]$. What is the true parameter value, $\Psi(P_0)$, for this data generating process?

(c) Now consider the parameter, $\Psi^2(P) = E_P E_P[Y|A = 1, W]$ or the true treatment specific mean for our non-parametric model. What is the true parameter value for the data generating process in part (b)?

(d) Run a simulation where you take 1000 draws of $n = 1000$ from the data generating process in part (b). Using your NPMLE from part (a) and its influence curve to form 95% Wald confidence intervals, check coverage of the true Ψ and Ψ^2 for each draw. Report the coverage percentage for each parameter. Considering that estimating Ψ is often used to estimate Ψ^2, comment on your results.

Problem 3

Bonus question: What is the remainder term for your estimator in part (a)? What remainder term conditions need to be satisfied for the estimator to be asymptotically linear?

We assume the NPMLE is well-behaved in the sense that it estimates the parameters in the second-order remainder at a rate faster than $n^{-1/4}$ (i.e. consistent at rate $< n^{-1/4}$). Then, $R_2(\mathbf{P}_n, P_0) = o_P(n^{-1/2})$ and $\Psi(\mathbf{P}_n) - \Psi(P_0) = (P_n - P_0)D^*(\mathbf{P}_n) + o_P(n^{-1/2})$. We had to make this assumption to derive the efficient influece curve for the NPMLE so the remainder term in part (a) is

Collaborators & Resources

Tommy Carpenito, Yue You

Jonathan's lab3sol