

PH 716 Applied Survival Analysis

Part VI: Diagnostics of Cox PH Models

Zhiyang Zhou (zhou67@uwm.edu, zhiyanggeezhou.github.io)

2024/Mar/31 22:09:04

Types of residuals

- Cox-Snell residuals: assessing the overall fit of the final model
- Martingale residuals: determining the functional form of a covariate included in the model
- Deviance residuals: detecting outliers
- Schoenfeld residuals: checking the appropriateness of the PH assumption

Cox-Snell residuals

- Cox-Snell residuals: $r_{i,CS} = \hat{\Lambda}_{T_i}(\tilde{T}_i)$
 - $\hat{\Lambda}_{T_i}(\cdot)$: estimated $\Lambda_{T_i}(\cdot)$ given by the Cox PH model
 - (For uncensored subjects) $r_{i,CS} \stackrel{\text{iid}}{\approx} \exp(1)$
 - (For all subjects) $\hat{\Lambda}_{H_i,NA}(\tilde{t}_i) \approx \tilde{t}_i$
 - * $H_i = \hat{\Lambda}_{T_i}(T_i)$
 - Note that $r_{i,CS} \neq H_i$ and instead $r_{i,CS} = \min\{H_i, \hat{\Lambda}_{T_i}(C_i)\}$
 - * $\hat{\Lambda}_{H_i,NA}(\cdot)$: NA estimator of $\Lambda_{H_i}(\cdot)$ based on right-censored $\{(r_{i,CS}, \Delta_i) : i = 1, \dots, n\}$
- Cox-Snell residual plot
 - (For uncensored subjects) compare $r_{i,CS}$ to $\exp(1)$ samples via Q-Q plot
 - (For all subjects) plot $\hat{\Lambda}_{H_i,NA}(\tilde{t}_i)$ against \tilde{t}_i
 - Used to diagnose poor model fit
 - No insight into how model assumptions are violated
- Theoretical notes
 - Inverse cdf theorem: arbitrary r.v. X with cdf $F_X(x) = \Pr(X \leq x) \Rightarrow F_X(X) \sim U(0, 1)$
 - It follows that $T_i \stackrel{\text{independent}}{\sim} S_{T_i}(\cdot) \Rightarrow S_{T_i}(T_i) \stackrel{\text{iid}}{\sim} U(0, 1) \Rightarrow \Lambda_{T_i}(T_i) = -\ln S_{T_i}(T_i) \stackrel{\text{iid}}{\sim} \exp(1)$
 - $\{(r_{i,CS}, \Delta_i) : i = 1, \dots, n\}$ is a right-censored dataset
 - * $r_{i,CS} = \min(H_i, \hat{\Lambda}_{T_i}(C_i)) \Leftarrow \tilde{T}_i = \min(T_i, C_i)$ and monotonically ascending $\hat{\Lambda}_{T_i}(\cdot)$
 - $H_i = \hat{\Lambda}_{T_i}(T_i) \approx \Lambda_{T_i}(T_i) \Rightarrow H_i \approx \exp(1) \Rightarrow \Lambda_{H_i}(\tilde{t}_i) \approx \tilde{t}_i$

Ex. 6.1 [KM, Example 11.1]

- This multi-center acute leukemia study consists of 137 patients with acute myelocytic leukemia (AML) or acute lymphoblastic leukemia (ALL) aged 7 to 52 from March 1, 1984 to June 30, 1989 at four institutions.
- The disease-free survival time (**t2**) on study is defined as time (in days) to relapse or death
- **d3** is the disease free survival indicator: 1 - Dead or Relapsed, 0 - Alive Disease Free.
- Focus on effects of the following 9 covariates on disease-free survival:

- z1: Patient age in years.
- z2: Donor age in years.
- z3: Patient sex: 1 - Male, 0 - Female.
- z4: Donor sex: 1 - Male, 0 - Female.
- z5: Patient Cytomegalovirus (CMV) status: 1 - CMV positive, 0 - CMV negative.
- z6: Donor CMV status: 1 - CMV positive, 0 - CMV negative.
- z7: Waiting time to transplant in days.
- z8: French–American–British classification (FAB): 1 - FAB Grade 4 or 5 and AML, 0 - otherwise.
- z10: Methotrexate (MTX): used as a Graft-Versus-Host-Prophylactic 1 - Yes, 0 - No.

```
options(digits=4)
library(survival)
# model fitting
data.ex55 = read.csv("bmt.csv")
fit.ex55 <- coxph(Surv(t2,d3) ~ z1+z2+z3+z4+z5+z6+z7+z8+z10, data=data.ex55)

# Cox-Snell residual
r.cs = data.ex55$d3$residuals(fit.ex55, type='martingale') # Cox-Snell

# Cox-Snell residual plot
set.seed(2024)
exp.rnd = rexp(10000)
qqplot(
  x = exp.rnd, y = r.cs[as.logical(data.ex55$d3)],
  xlab = "Theoretical Quantiles", ylab = "Sample Quantiles"
)
qqline(r.cs[as.logical(data.ex55$d3)], distribution = qexp)
# Or
cum.haz.r.cs <- basehaz(coxph(Surv(r.cs, d3)~1, data=data.ex55), centered = FALSE)
plot(
  x=cum.haz.r.cs[,2], y=cum.haz.r.cs[,1],
  xlab='t', ylab='Cumulative hazard of r.cs'
)
abline(a=0,b=1,col='red')
```

Martingale residuals

- Martingale residuals: $r_{i,M} = \Delta_i - r_{i,CS}$
- To explore the proper functional form of the j_0 th covariate, say $f_{j_0}(\cdot)$
 1. Fit a Cox PH model without the j th covariate $\lambda_{T_i}(t) = \lambda_0(t) \exp\{\sum_{j \neq j_0} f_j(x_{ij})\beta_j\}$
 - $f_j(\cdot)$: known proper functional form of the j th covariate
 2. Compute martingale residuals $r_{i,M}$ for the above model
 3. Scatterplot of $r_{i,M}$ against x_{ij_0} with a fitted locally estimated scatterplot smoothing (loess) curve
 - $f_{j_0}(\cdot)$ indicated by the loess curve
 4. Update the model by adding $f_{j_0}(x_{ij_0})$ and check the scatterplot of updated $r_{i,M}$ against x_{ij_0} with a fitted loess line
 - No further transformation needed If the new loess curve lies at the x-axis
- Theoretical notes:
 - loess: to predict $f(x_0)$ by an weighted average $\hat{f}(x_0) = \sum_{i=1}^n w_i y_i$, where w_i relies on the distance between x_i and x_0 ; a closer x_i enjoys a higher w_i .
 - Why is the residual bearing such a name?
 - * Martingale: a stochastic process $M(t)$ such that $E\{M(t)\} = 0$ and $E\{M(t) \mid M(s)\} = M(s)$

- for all $s < t$
- * $r_{i,M}$ obtained by evaluating a martingale at \tilde{t}_i
- Why would the martingale residuals reveal the correct functional forms of covariates?
 - * Because $E(r_{i,M}) \approx (n_D/n)\{f_1(x_{i1}) - \text{constant}\}$ [KM, pp. 362]
 - n_D/n : the ratio of total number of events to total number of subjects
- Zero-sum of martingale residuals: $\sum_i r_{i,M} = 0$
 - * Specific for Cox PH model with the Breslow estimator for the baseline cumulative hazard
 - * Proof: $\sum_i r_{i,CS} = \sum_i \sum_{k:t_k \leq \tilde{t}_i} \frac{d_k \exp(\sum_j x_{ij} \hat{\beta}_j)}{\sum_{\ell \in \mathcal{R}(t_k)} \exp(\sum_j x_{\ell j} \hat{\beta}_j)} = \sum_k \sum_{i \in \mathcal{R}(t_k)} \frac{d_k \exp(\sum_j x_{ij} \hat{\beta}_j)}{\sum_{\ell \in \mathcal{R}(t_k)} \exp(\sum_j x_{\ell j} \hat{\beta}_j)} = \sum_k d_k = \sum_i \delta_i$

Revisit Ex. 6.1

```
options(digits=4)
library(survival)
# [DM, pp. 208] a function to add the smooth curve and confidence limits
smoothSEcurve <- function(yy, xx) {
  # use after a call to "plot"
  # fit a lowess curve and 95% confidence interval curve
  # make list of x values
  xx.list <- min(xx) + ((0:100)/100)*(max(xx) - min(xx))
  # Then fit loess function through the points (xx, yy)
  # at the listed values
  yy.xx <- predict(loess(yy ~ xx, span = 1), se=T, newdata=data.frame(xx=xx.list))
  lines(yy.xx$fit ~ xx.list, lwd=2)
  lines(yy.xx$fit -
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
  lines(yy.xx$fit +
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
}

# model fitting without z1
data.ex55 = read.csv("bmt.csv")
fit.ex55 <- coxph(Surv(t2,d3) ~ z2+z3+z4+z5+z6+z7+z8+z10, data=data.ex55, ties = 'exact')

# Martingale residual plot (for the model without z1) vs. z1
r.m = residuals(fit.ex55, type='martingale')
sum(r.m)
plot(
  x=data.ex55$z1, y=r.m,
  main = 'Martingale residuals \n (for the model without z1) \n versus z1')
smoothSEcurve(r.m, data.ex55$z1) # indicating a cubic function?

# model fitting with a cubic function of z1
fit.ex55.1 <- coxph(Surv(t2,d3) ~ z1+I(z1^2)+I(z1^3)+z2+z3+z4+z5+z6+z7+z8+z10, data=data.ex55)

# Martingale residual plot (for the model with a cubic function of z1) vs. z1
r.m.1 = residuals(fit.ex55.1, type='martingale')
plot(
  x=data.ex55$z1, y=r.m.1,
  main = 'Martingale residual \n (for the model with a cubic function of z1) \n versus z1')
smoothSEcurve(r.m.1, data.ex55$z1)
```

Deviance residuals

- Outlier: an observation for which the outcome is not sufficiently well predicted by the fitted model
- Deviance residuals: $r_{i,D} = \text{sign}(r_{i,M}) \sqrt{-2\{r_{i,M} + \delta_i \ln(\delta_i - r_{i,M})\}}$
 - Symmetrically distributed with expected value 0 (if the fitted model is correct); de-skewed/transformed martingale residuals
 - * $r_{i,D} = 0 \Leftrightarrow r_{i,M} = 0$
 - * Inflating $r_{i,D}$ when $r_{i,M}$ is close to 1
 - * Shrinking large negative $r_{i,M}$
 - Analogous to the deviance in GLMs
- Detecting outliers: plotting $r_{i,D}$ against $\sum_{j=1}^p x_{ij}\hat{\beta}_j$ (called linear predictors or risk scores)
 - With moderate (or less) censoring, this plot should look like randomly-distributed noise without discernible pattern
 - Large absolute values of deviance residuals indicating observations that are poorly explained by the model, potentially pointing to outliers or influential points
 - * 95% of absolute deviance residuals ≤ 2
 - * 99.7% of absolute deviance residuals ≤ 3

Revisit Ex. 6.1

```
options(digits=4)
library(survival)
# [DM, pp. 208] a function to add the smooth curve and confidence limits
smoothSEcurve <- function(yy, xx) {
  # use after a call to "plot"
  # fit a lowess curve and 95% confidence interval curve
  # make list of x values
  xx.list <- min(xx) + ((0:100)/100)*(max(xx) - min(xx))
  # Then fit loess function through the points (xx, yy)
  # at the listed values
  yy.xx <- predict(loess(yy ~ xx, span = 1), se=T, newdata=data.frame(xx=xx.list))
  lines(yy.xx$fit ~ xx.list, lwd=2)
  lines(yy.xx$fit -
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
  lines(yy.xx$fit +
  qt(0.975, yy.xx$df)*yy.xx$se.fit ~ xx.list, lty=2)
}

# model fitting
fit.ex55.1 <- coxph(
  Surv(t2,d3) ~ z1+I(z1^2)+I(z1^3)+z2+z3+z4+z5+z6+z7+z8+z10,
  data=data.ex55,
  x = T
)

# Two ways to calculate linear predictors
risk.score.1 = fit.ex55.1$x %>% coef(fit.ex55.1)
risk.score.2 = fit.ex55.1$linear.predictors
sum((risk.score.1-risk.score.2)^2) # seems distinct?

# Deviance residual plot vs. risk scores
r.d = residuals(fit.ex55.1, type='deviance')
```

```

plot(
  x=risk.score.1, y=r.d,
  main = 'Deviance residuals \n versus risk scores')
smoothSEcurve(yy=r.d, xx=risk.score.1)
abline(a=2,b=0,col='red')
abline(a=-2,b=0,col='red')
abline(a=3,b=0,col='red')
abline(a=-3,b=0,col='red')

(1:nrow(data.ex55))[abs(r.d) > 2] # Potential outliers
sum(abs(r.d) > 2)/nrow(data.ex55) # percent of r.d over 2
sum(abs(r.d) > 3)/nrow(data.ex55) # percent of r.d over 3

```

Schoenfeld residuals

- Schoenfeld residuals: for UNCENSORED subject i and the j th covariate,

$$r_{ij,S} = x_{ij} - \bar{x}_{.j}$$

- $\bar{x}_{.j} = \sum_{k \in \text{uncensored subjects}} w_{kj} z_{kj}$ with weights $w_{kj} = \frac{\exp(\sum_{j=1}^p x_{kj} \beta_j)}{\sum_{\ell \in \mathcal{R}(\tilde{t}_k)} \exp(\sum_{j=1}^p x_{\ell j} \beta_j)}$
- [DM, Sec. 7.2.2] Schoenfeld residuals are components of the score function $\Rightarrow \sum_{i \in \text{uncensored subjects}} r_{ij,S} = 0$ for each j
- Scaled Schoenfeld residuals

$$r_{ij,S}^* = d \cdot r_{ij,S} \cdot \text{var}(\hat{\beta}_j)$$
 - d : total number of events
 - If the hazard ratio is constant over time, then $E(r_{ij,S}^*) + \beta_j$ is time-invariant
- Investigating the PH assumption
 - Plotting $r_{ij,S}$ versus the covariate x_{ij} for the j covariate
 - * Points centered at zero if the PH assumption holds
 - * Inconvenient to be implemented in R
 - Instead, checking the plot of $r_{ij,S}^* + \hat{\beta}_j$ vs. t
 - * Points without a time tendency if the PH assumption holds
 - Score test (`survival::cox.zph`)

Revisit Ex. 6.1

```

options(digits=4)
library(survival)

# model fitting
fit.ex55.1 <- coxph(
  Surv(t2,d3) ~ z1+I(z1^2)+I(z1^3)+z2+z3+z4+z5+z6+z7+z8+z10,
  data=data.ex55,
  x=T
)

# (unscaled) Schoenfeld residuals
r.s.unscaled = residuals(fit.ex55.1, type='schoenfeld')

# (scaled) Schoenfeld residual plot
plot(cox.zph(fit.ex55.1, transform="identity", terms=F, global=F))

```

```
# Score test  
cox.zph(fit.ex55.1, transform="identity", terms=F, global=F)
```