Applied Survival Analysis Using R
Chapter 7: Model Diagnostics

Qi Guo

Department of Mathematical Sciences
The University of Texas at Dallas

April, 15 2019

1 Assessing Goodness of Fit Using Residuals

2 Case Deletion Residuals

3 Checking the Proportion Hazards Assumption

4 Schoenfeld Residuals

The use of residuals

- The *residuals* are plotted versus some *quantity*(STAT 6337: Advanced Statistical Method I, the first two chapters), such as a covariate value, and the observed pattern is used to diagnose possible problems with the fitted model.
- The *residuals* have the additional property of not only indicating problems but also suggesting *remedies*, that means suggest an alternative model that fits the data better.
- Many of these residuals have been generalized to survival analysis. And survival data *evolves over time*, and requires special assumptions such as proportional hazard.

Martingale and Deviance Residuals

- An important tool for *assessing the goodness of fit* of a model is to *compare the censoring indicator* (0 for censored, 1 for death) for each subject to the *expected* value of that indicator under the *proportional hazards Cox model*.
- If there are *no time dependent covariates*(Chapter 8) and if the survival times are right-censored.*Martingale Residuals*

$$m_i = \delta_i - \hat{H}_0(T_i)exp(z_i^{'}\hat{\beta}) \tag{1}$$

- The residual is essentially the *difference* between the observed value (1 or 0) of the censoring indicator and its expected value under a particular Cox model.
- But the sum of squares of Martingale residuals cannot be used as a measure of goodness of fit, so "**deviance**" residual is an alternative way does have this property.

$$d_i = sign(m_i) \cdot \{-2 \cdot [m_i + \delta_i \log(\delta_i - m_i)]\}^{1/2} \tag{2}$$

Difference

1. Deviance Residuals:
   - These residuals are *symmetrically* distributed with expected value 0 (if the fitted model is correct)
   - The sum of squares of these residuals is the value of the likelihood ratio test.
2. Martingale Residuals:
   - These residuals have the property of showing us the *functional form of a covariate*.

- Conclusion: For this reason, in practice, the *martingale residuals are more useful*.

Example

Example 7.1

Consider the “`pharmacoSmoking`” dataset,and a fit of the “`null`” Cox proportional hazards model. A null model is one with no fitted covariates. plot Martingale Residuals against continuous predictors to get a preliminary assessment of which of these predictors should be in the model, and what form they should take

- Read in the data and truncate the variable “`priorAttempts`” at 20.

```
> pharmacoSmoking <- read.csv("PharmacoSmoking.csv")
> priorAttemptsT <- priorAttempts
> priorAttemptsT[priorAttempts > 20] <- 20
```

- Fit the null model and obtain these residuals as follows:

```
> library(survival)
> result.0.coxph <- coxph(Surv(ttr, relapse) ~ 1)
> rr.0 <- residuals(result.0.coxph, type="martingale")
```

Example

- Plot *martingale residuals* versus `age`, `prior attempts at quitting`, and `longest prior period without smoking`.
- Plot these null model residuals versus each of these variables and also versus *log transformations* of these variables.

```
> par(mfrow=c(3,2))
> plot(rr.0 ~ age)
> smoothSEcurve(rr.0, age)
> title("Martingale residuals versus age")

> logAge <- log(age)
> plot(rr.0 ~ logAge)
> smoothSEcurve(rr.0, logAge)
> title("Martingale residuals versus log age")

> plot(rr.0 ~ priorAttemptsT)
> smoothSEcurve(rr.0, priorAttemptsT)
> title("Martingale residuals versus prior attempts")
```

Example

```
> logPriorAttemptsT <- log(priorAttemptsT + 1)
> plot(rr.0 ~ logPriorAttemptsT)
> smoothSEcurve(rr.0, logPriorAttemptsT)
> title("Martingale residuals versus log prior attempts")

> plot(rr.0 ~ longestNoSmoke)
> smoothSEcurve(rr.0, longestNoSmoke)
> title("Martingale residuals versus longest period without smoking")

> logLongestNoSmoke <- log(longestNoSmoke+1)
> plot(rr.0 ~ logLongestNoSmoke)
> smoothSEcurve(rr.0, logLongestNoSmoke)
> title("Martingale residuals versus log of longest period without sm
```

- In this figure, the three plots on the *left* are for *untransformed* variables, and the three on the *right* are for *log transformations* of these variables.

Example

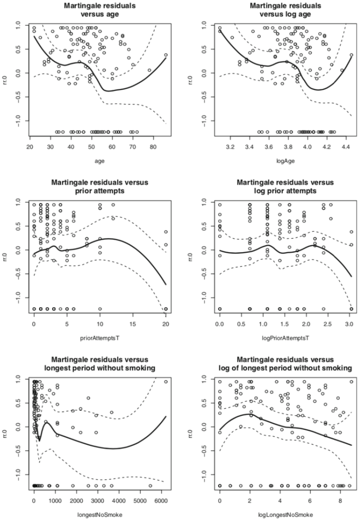


Fig. 7.1 Martingale residuals from a null model fit to the pharmacoSmoking data, plotted versus three continuous predictors and one log-transformed predictor

Case Deletion Residuals

- Some subjects may have an *especially large influence* on the parameter estimates and may cause problems, so we have to identify them.
- *Case deletion residuals* (also called “jackknife residuals”) serve this purpose.

Case Deletion Residuals

For each subject, a case deletion residual is the *difference* in the value of the coefficient using all of the data and its value when that *subject is deleted* from the data set.

```
> result.coxph <- coxph(Surv(ttr, relapse) ~ grp + employment
+ age)
> coef.all <- result.coxph$coef[4]
> coef.all
        age
-0.03528934
```

Delete

- For each subject in turn ("`n.obs`" subjects in all), we delete the *i*th subject from the survival time "`tt`", censoring indicator "`relapse`", and covariates "`grp`", `employment`", and "`age`"
- Fit a Cox model to this reduced data set. The results for the *i*th subject go into "`result.coxph.i`"
- Extract the age coefficient(4th) into "`coef.i`" and compute the jackknife residual as the difference of "`coef.i`" and "`coef.all`":

```
> n.obs <- length(ttr)
> jkbeta.vec <- rep(NA, n.obs)
> for (i in 1:n.obs) {
tt.i <- ttr[-i]
delta.i <- relapse[-i]
grp.i <- grp[-i]
employment.i <- employment[-i]
age.i <- age[-i]
result.coxph.i <- coxph(Surv(tt.i, delta.i) ~ grp.i +
employment.i + age.i)
coef.i <- result.coxph.i$coef[4]
jkbeta.vec[i] <- (coef.all - coef.i)}
```

Plot

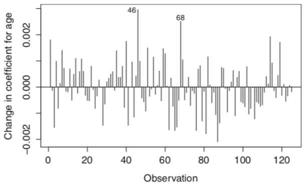
- Plot these residuals versus the patient id’s, which we place in the vector “index.obs”, “type=’h’” causes the residuals to be plotted as spikes. “abline(h=0)” plots a horizontal line through 0.

```
> index.obs <- 1:n.obs
> plot(jkbeta.vec ~ index.obs, type="h",xlab="Observation",
ylab="Change in coefficient for age", cex.axis=1.3, cex.lab=1.3)
> abline(h=0)
```

- The “identify” function allows us to identify the index numbers of select patients by manually selecting them with a mouse

```
> identify(jkbeta.vec ~ index.obs)
```

Conclusion



- Conclusion: No single patient changes the estimate of the "`age`" coefficient by more than 0.003, which is less than 10% of the value of that coefficient. and patients 46 and 68 have the most influence over the parameter estimate for age.
- A more convenient way to obtain case deletion residuals is using the "`residuals`" function with "`type = 'dfbeta'`".

debeta residuals

```
> resid.dfbeta <- residuals(result.coxph, type="dfbeta")
> n.obs <- length(ttr)
> index.obs <- 1:n.obs
> plot(resid.dfbeta[,4] ~ index.obs, type="h",xlab="Observation",
ylab="Change in coefficient")
> abline(h=0)
> identify(resid.dfbeta[,4] ~ index.obs)
```

- The resulting `dfbeta` residuals plot (not shown) is nearly *identical*, it has the advantage that it is slightly easier to use it to produce multiple plots for all of the coefficients.

Log Cumulative Hazard Plots

- log-log transformation:
  - Under the assumption, we have

    $$S_1(t) = [S_0(t)]^{exp(\beta)} \tag{3}$$

    where $exp(\beta)$ is the proportional hazards constant.
  - Take logs both sides

    $$\log[S_1(t)] = exp(\beta) \cdot \log[S_0(t)] \tag{4}$$

  - The survival functions are less than 1, so log will cause negative.

    $$\log\{-\log[S_1(t)]\} = \beta + \log\{-\log[S_0(t)]\} \tag{5}$$

  where the function $g(\mu) = \log\{-\log(\mu)\}$ is called a *complementary log-log transformation*.
- Plot $g[S_1(t)]$ and $g[S_0(t)]$ versus t or $\log(t)$ will yield *two parallel curves* separated by β if the proportional hazards assumption is correct.

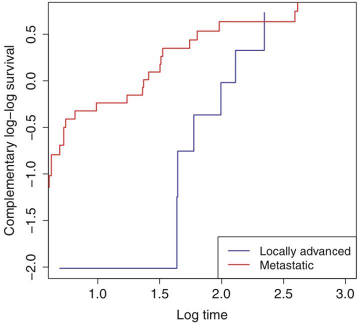
Example

- Illustrate this with the pancreatic cancer data from Chapter 3.

```
# Plot the LA(Locally advanced)
> result.surv.LA <- survfit(Surv(pfs.month) ~ stage,
subset=stage == "LA")
> time.LA <- result.surv.LA$time
> surv.LA <- result.surv.LA$surv
> cloglog.LA <- log(-log(surv.LA))
> logtime.LA <- log(time.LA)

# Plot the M(Metastatic)
> result.surv.M <- survfit(Surv(pfs.month) ~ stage,
subset=stage == "M")
> time.M <- result.surv.M$time
> surv.M <- result.surv.M$surv
> cloglog.M <- log(-log(surv.M))
> logtime.M <- log(time.M)
> plot(cloglog.LA ~ logtime.LA, type="s", col="blue", lwd=2)
> lines(cloglog.M ~ logtime.M, col="red", lwd=2, type="s")
> legend("bottomright", legend=c("Locally advanced",
"Metastatic"), col=c("blue","red"), lwd=2)
```

Example



Complementary log-log plot for the two pancreatic cancer groups

- Conclusion: The curves are clearly *not parallel*. However, one problem with this approach is that we don't have a clear way to assess *statistical significance*. This is a critical issue here due to the *small sample size*, particularly in the locally advanced group.

Recall

- *The Schoenfeld residuals* are the individual terms of *the score function*, and each term is the observed value of the covariate for patient i minus the expected value $E(Z_i) = \bar{z}(t_i)$, which is *a weighted sum*, with weights given by $p_k(\beta)$, of the covariate values for subjects at risk at that time.
- Recall:
  - The partial log-likelihood function

$$\ell(\beta) = \sum_{i \in D} \left\{ \log(\psi_i) - \log\left(\sum_{k \in R_i} \psi_k \right) \right\} = \sum_{i \in D} \left\{ z_i \beta - \log\left(\sum_{k \in R_i} e^{z_k \beta} \right) \right\} \tag{6}$$

  - The score function:

$$\ell^{'}(\beta) = \sum_{i \in D} \left\{ z_i - \sum_{k \in R_i} z_k \cdot p(\beta, z_k) \right\} where\ p(\beta, z_k) = \frac{e^{z_k \beta}}{\sum_{j \in R_k} e^{z_j \beta}} \tag{7}$$

Example

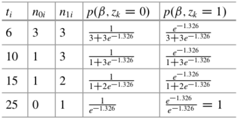
- For an estimate $\hat{\beta}$ the residual for the ith failure time is:

$$\hat{r}_i = z_i - \sum_{k \in R_i} z_k \cdot p(\hat{\beta}, z_k) = z_i - \bar{z}(t_i) \tag{8}$$

- Example:

```
> tt <- c(6, 7, 10, 15, 19, 25)
> delta <- c(1, 0, 1, 1, 0, 1)
> trt <- c(0, 0, 1, 0, 1, 1)
> result.coxph <- coxph(Surv(tt, delta) ~ trt) > result.coxph$coef
[1] trt -1.326
```

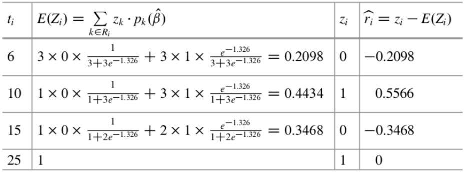
- We see that $\hat{\beta} = 1.326$. To compute the Schoenfeld residuals, we first compute the weights as follows:



t_i	n_{0i}	n_{1i}	$p(\beta, z_k = 0)$	$p(\beta, z_k = 1)$
6	3	3	$\frac{1}{3+3e^{-1.326}}$	$\frac{e^{-1.326}}{3+3e^{-1.326}}$
10	1	3	$\frac{1}{1+3e^{-1.326}}$	$\frac{e^{-1.326}}{1+3e^{-1.326}}$
15	1	2	$\frac{1}{1+2e^{-1.326}}$	$\frac{e^{-1.326}}{1+2e^{-1.326}}$
25	0	1	$\frac{1}{e^{-1.326}}$	$\frac{e^{-1.326}}{e^{-1.326}} = 1$

Example

- Compute the expected values, and then the residuals:



t_i	$E(Z_i) = \sum_{k \in R_i} z_k \cdot p_k(\hat{\beta})$	z_i	$\widehat{r_i} = z_i - E(Z_i)$
6	$3 \times 0 \times \frac{1}{3+3e^{-1.326}} + 3 \times 1 \times \frac{e^{-1.326}}{3+3e^{-1.326}} = 0.2098$	0	-0.2098
10	$1 \times 0 \times \frac{1}{1+3e^{-1.326}} + 3 \times 1 \times \frac{e^{-1.326}}{1+3e^{-1.326}} = 0.4434$	1	0.5566
15	$1 \times 0 \times \frac{1}{1+2e^{-1.326}} + 2 \times 1 \times \frac{e^{-1.326}}{1+2e^{-1.326}} = 0.3468$	0	-0.3468
25	1	1	0

- In R:

```
> residuals(result.coxph, type="schoenfeld")
[1]   6        10           15         25
-0.2098004 0.5566351 -0.3468347 0.0000000
```

The scaled residual

- Gramsch and Therneau proposed scaling each residual by an estimate of its variance. This *scaled residual* conveniently approximated as follows:

$$r_i^\star = r_i \cdot d \cdot var(\hat{\beta}) \tag{9}$$

where d is the total number of deaths, and $var(\hat{\beta})$ is the variance of the parameter estimate.

```
> resid.unscaled <- residuals(result.coxph, type= ``schoenfeld'')
> resid.scaled <- resid.unscaled⋆result.coxph$var⋆sum(delta)
> resid.unscaled
[1]      6         10         15         25
   -0.2098004 0.5566351 -0.3468347 0.0000000
> resid.scaled
[1] -1.313064 3.483776 -2.170712 0.000000
```

The scaled residuals

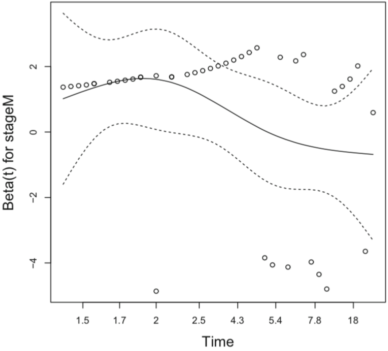
- An approximate estimate of $\beta(t)$ may be obtained by adding the estimate $\hat{\beta}$ from the Cox proportional hazards model to the standardized residuals.

```
> resid.scaled + result.coxph$coef
[1] -2.639193 2.157647 -3.496841 -1.326129
# or using ``cox.zph'' function
> resid.sch <- cox.zph(result.coxph)
> resid.sch$y
       trt
6  -2.639193
10  2.157647
15 -3.496841
25 -1.326129
```

Example

- Using this residuals to the pancreatic data, compute the *scaled Schoenfeld residuals* and plot

```
> result.coxph <- coxph(Surv(pfs.month) ~ stage)
#The ``transform'' option specifies that the time axis is scaled to
> result.sch.resid <- cox.zph(result.coxph, transform="km")]
> plot(result.sch.resid)
```



Example

- The shape of the *smoothed(loess) curve* is an estimate of the difference parameter as a function of time.
- The hypothesis test for a constant β

```
> result.sch.resid
         rho   chisq    p
stageM -0.328  3.86   0.0496
```

- Alternatively, variations of this test may be obtained by plotting $\beta(t)$versus time or versus other transformations of time, and the "`cox.zph`" function offers:
  - "`rank`" option: the time axis is ordered by the ranks of the times.
  - "`identity`" option: the time variable is untransformed.

```
> cox.zph(result.coxph, transform="rank")
         rho    chisq    p
stageM  -0.33    3.89   0.0486
> cox.zph(result.coxph, transform="identity")
         rho    chisq    p
stageM  -0.197   1.39   0.239
```