7. Model building and diagnostics

§7.1. Model building with hazard proportional models

A modeling building procedure consists of two components: 1) a searching procedure and 2) a model selection criterion. The classical searching procedures and model selection criteria for ordinary regression analysis can also be applied to model building with hazard regression models.

- **Searching procedures:**
  - Forward procedure;
  - Backward procedure;
  - Stepwise forward/backward procedure.

- **Selection Criteria:**
  \[
  \text{AIC: } -2 \log L(\hat{\beta}) + 2p, 
  \]
where $\hat{\beta}$ is the MLE of the coefficient vector $\beta$ and $p$ is the number of components of $\beta$.

BIC: $-2 \log L(\hat{\beta}) + p \log n$,
where $n$ is the sample size.

CV (cross validation), etc..

\textbf{Forward Procedure in detail}

Call features of primary concern primary features, others adjusting features.

\textbf{Step 0} : Fit the model with primary features. Compute its AIC.

\textbf{Step 1} : Take the model including the primary features as the original model. Consider models obtained by adding one additional adjusting feature to the original model. Among all such models, choose the one with the largest likelihood and compute the AIC of this model. If the AIC is smaller than that of the original model, retain the feature corresponding to the smallest AIC in the model and continue.

\textbf{Step 2} : Update the original model with the newly added adjusting feature, repeat Step 1, until no more features can be added.
• Bone Marrow Transplant example (cont.)

```r
bmt=read.table("bmt.txt")
attach(bmt)

# Group indicator
g1=bmt$g
g1[g1!=2]=0
g1[g1==2]=1

g2=bmt$g
g2[g2!=3]=0
g2[g2==3]=1

detach(bmt)
bmt=data.frame(bmt,g1,g2)
attach(bmt)

Y=Surv(T2,d3)
X=bmt[,c(13:20,22)]
Z1Z2=Z1*Z2
Z3Z4=Z3*Z4
Z5Z6=Z5*Z6
X=data.frame(cbind(X,Z1Z2,Z3Z4,Z5Z6))
p=dim(X)[2]
c.set = 1:p

fit.0=coxph(Y~g1+g2,method="breslow")
AIC0=-2*fit.0$loglik[2]+2*2
```
AIC=NULL
x.seq =NULL
X.m = data.frame(cbind(g1,g2))
for (i in 1:p) {
  p.i = length(c.set)
  jmin = 1
  AIC.min = 10^(8)
  for (j in 1:p.i) {
    X.tmp = data.frame(cbind(X.m,X[,c.set[j]])
    fit.tmp=coxph(Y~.,X.tmp,method="breslow")
    AIC.tmp=-2*fit.tmp$loglik[2]+2*dim(X.tmp)[2]
    if (AIC.tmp < AIC.min) {
      jmin = j
      AIC.min = AIC.tmp
    }
  }
  x.seq[i] = c.set[jmin]
  AIC[i] = AIC.min
  X.m = data.frame( cbind(X.m, X[, x.seq[i]]) )
  c.set=c.set[-jmin]
}
AIC0
  737.2883
data.frame(x.seq, AIC)

  x.seq  AIC
  1     8 731.0202
  2    10 729.9720
  3     2 728.9400
  4     1 725.9818
  5     9 726.5797
fit.final = coxph(Y~g1+g2+Z1+Z2+Z1Z2+Z8,data=bmt,method="breslow")
summary(fit.final)

• An introduction to more advanced model selection procedures

Greedy nature of stepwise procedures.
Penalized likelihood approach: minimizing
\[-2 \log L(\beta) + \lambda \sum_j |\beta_j|.

Choosing \(\lambda\) by multi-fold cross-validation.
§7.2. Cox-Snell residuals for assessing the fit of Cox model

- **Cox-Snell residuals**

  \[ r_j = \hat{H}_0(T_j) \exp\left( \sum_{k=1}^{p} b_k x_{jk} \right), \quad j = 1, \ldots, n. \]

  \[ b = (b_1, \ldots, b_p)^t: \text{MLE of } \beta. \]

  \[ \hat{H}_0(t): \text{MLE of baseline cumulative hazard function.} \]

  Note: If model is correct,

  \[ r_j = \hat{H}(T_j|\mathbf{x}_j) \]

  \[ = - \ln[\hat{S}(T_j|\mathbf{x}_j)] \]

  \[ \sim \mathcal{E}(\lambda = 1), \]

  since \( u = S(T_j|\mathbf{x}_j) \sim \mathcal{U}[0, 1], \)

  and \( - \ln u \sim \mathcal{E}(\lambda = 1). \)
• Checking the fit of Cox model using Cox-Snell residuals

Since the cumulative hazard function of $\mathcal{E}(\lambda = 1)$ is linear. This gives rise to the following procedure for the checking the appropriateness of the fit.

i) Obtain the estimated cumulative hazard function of the Cox-Snell residuals. Denote the estimates by $\hat{H}_r(r)$.

ii) Plot $\hat{H}_r(r_j)$ against $r_j$. Check whether the plot is linear through the origin with a slope 1. If yes, the fitted model is adequate. Any departure from this indicates a lack of fit.

Note, when there is only right censoring and time-independent covariates, the Cox-Snell residual is closely related to the martingale residual to be defined later. The martingale residual
can be obtained by the R function `residuals`.

• Bone Marrow Transplant example (cont.)

```r
diag.fit=coxph(Surv(T2,d3)~g1+g2+Z1+Z2+Z1Z2+Z7+Z8+Z10, data=bmt)
m.resi=residuals(diag.fit, type="martingale")
cs.resi=bmt$d3-m.resi
r.surv1=survfit(Surv(cs.resi,bmt$d3)~1, type="fleming-harrington")
plot(r.surv1$time, -log(r.surv1$surv))
lines(c(0,3),c(0,3))
diag.fit=coxph(Surv(T2,d3)~g1+g2+Z1+Z2+Z1Z2+Z7+Z8+strata(Z10),data=bmt)
m.resi=residuals(diag.fit, type="martingale")
cs.resi=bmt$d3-m.resi
r.surv2=survfit(Surv(cs.resi,bmt$d3)~1, type="fleming-harrington")
plot(r.surv2$time, -log(r.surv2$surv))
lines(c(0,3),c(0,3))
par(mfrow=c(1,2))
plot(r.surv1$time, -log(r.surv1$surv))
lines(c(0,3),c(0,3))
plot(r.surv2$time, -log(r.surv2$surv))
lines(c(0,3),c(0,3))
```
\section*{7.3 Martingale residuals for determining function form of covariates}

\begin{itemize}
  \item \textbf{Martingale residuals}
  \begin{itemize}
    \item The martingale residual is defined as
    \[
    \hat{M}_j = N_j(\infty) - \int_0^\infty Y_j(t) \exp[\hat{\beta}' \mathbf{x}_j(t)] d\hat{H}_0(t),
    \]
    where \(N_j(t) = I\{T_j \leq t, \delta_j = 1\}\).
    \end{itemize}
  \end{itemize}

\begin{itemize}
  \item When data is right censored and covariates are time-independent, it reduces to
    \[
    \hat{M}_j = \delta_j - r_j,
    \]
    where \(\delta_j\): indicator of censoring for individual \(j\),
    \(r_j\): Cox-Snell residual.
  \end{itemize}

\begin{itemize}
  \item \textbf{Plot of Martingale residuals against covariates}
  \begin{itemize}
    \item To determine the form of a covariate \(x\) in the model, fit a Cox model to all other covariates
  \end{itemize}
\end{itemize}
(excluding $x$) and obtain the martigale residuals $\hat{M}_j$. Then $\hat{M}_j$ is plotted against $x_j$.

1. If the plot is linear, no transformation is needed for $x$. If the plot is a curve, transformation for $x$ is in order.

2. A smoothed fit of the scatter diagram is used, which gives an indication of the transformation of $x$ if linearity does not hold.

3. If there appears to be a threshold, then, a discretized version of the covariate is indicated.
• **Example: BMT for Hodgkin and non-Hodgkin’s lymphoma.**

```
hodg_read.table("hodg.txt")

# V1: Graft type (1=allogenic, 2=autologous)
# V2: Disease type (1=Non Hodgkin lymphoma, 2=Hodgkins disease)
# V3: Time to death or relapse, days
# V4: Death/relapse indicator (0=alive, 1=dead)
# V5: Karnofsky score
# V6: Waiting time to transplant in months

attach(hodg)
X1=V1
X1[V1==2]=0
X2=c(rep(0,length(V2)))
X2[V2==2]=1
X3=X1*X2

hodg.fit=coxph(Surv(V3,V4)~X1+X2+X3+V5,data=hodg)
m.resi=residuals.coxph(hodg.fit, type="martingale")
plot(V6,m.resi)
lines(lowess(V6,mresi))
```

The plot in the next slide indicates that the waiting time could be coded as an indicator with shreshold roughly somewhere between 50 and 100.
Figure 7.2. Plot of martingale residual versus waiting time to transplant and LOWESS smooth
§7.4 Tests and graphical checks of the proportional hazard assumption

- **The R function `cox.zph`**

  The function takes a fitted `coxph` object as its argument. In the output, it provides

  i. The test statistics and *p*-values for the test of proportional hazard assumption for each fitted covariate.

  ii. A plot for each covariate. If the plot is horizontal, the proportional hazard can be assumed. Otherwise, the assumption is in doubt.

  R codes:

  ```
  coxph.fit=coxph(...)  
  zph.obj=cox.zph(coxph.fit)  
  print(zph.obj)  
  plot(zph.obj)
  ```
• Examples

1. Autologous and Allogeneic BMT for acute leukemia.

alloauto=read.table("alloauto.txt")
#V1: Time to death or relapse, months
#V2: Type of transplant (1=allogeneic, 2=autologous)
#V3: Leukemia-free survival indicator
# (0=alive without relapse, 1=dead or relapse)
aa=coxph(Surv(V1,V3)~V2, data=alloauto)
aa.zph=cox.zph(aa)
print(aa.zph)
plot(aa.zph)

Figure 7.3. Checking for Auto-Allo BMT.
2. Bone Marrow Transplant exampe (cont.)

bmt.fit=coxph(Surv(T2,d3)~g1+g2+Z1+Z2+Z1Z2+Z7+Z8+Z10,data=bmt)

bmt.zph=cox.zph(bmt.fit)
print(bmt.zph)
par(mfrow=c(4,2))
plot(bmt.zph)

<table>
<thead>
<tr>
<th>rho</th>
<th>chisq</th>
<th>p</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>-0.1110</td>
<td>1.2078</td>
</tr>
<tr>
<td>g2</td>
<td>-0.1758</td>
<td>2.8452</td>
</tr>
<tr>
<td>Z1</td>
<td>-0.0125</td>
<td>0.0166</td>
</tr>
<tr>
<td>Z2</td>
<td>-0.1997</td>
<td>3.8283</td>
</tr>
<tr>
<td>Z1Z2</td>
<td>0.1149</td>
<td>1.3954</td>
</tr>
<tr>
<td>Z7</td>
<td>-0.0499</td>
<td>0.2290</td>
</tr>
<tr>
<td>Z8</td>
<td>0.0833</td>
<td>0.5489</td>
</tr>
<tr>
<td>Z10</td>
<td>-0.2848</td>
<td>6.3139</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>NA</td>
<td>12.8730</td>
</tr>
</tbody>
</table>

The proportional hazard for \textbf{Z10} should be rejected.
Figure 7.4 Checking proportional hazard for Bone Marrow Transplant example
§7.5 Deviance residual for checking outliers

• Deviance residual

\[ D_j = \text{sign}(M_j)\{-2[M_j + \delta_j \log(\delta_j - M_j)]\}^{1/2}. \]

\( M_j \): martingale residuals, \( \delta_j \): indicators of censoring.

• Deviance plot

Plot \( D_j \) against \( s_j = \sum_{k=1}^{p} b_k x_{jk} \). Points standing out from the majority indicate that the corresponding individuals might be outliers.

```r
coxph.fit = coxph(...) 
d.resi = residuals(coxph.fit, 
                   type = "deviance") 
s = coxph.fit$linear.predictors 
plot(s, d.resi)
```
Example: BMT for Hodgkin and non-Hodgkin’s lymphoma (cont.)

X4=c(rep(0,length(V6)))
X4[V6>=84]=1
hodg.fit=coxph(Surv(V3,V4)~X1+X2+X3+X4+V5,data=hodg)
s=hodg.fit$linear.predictors
d.resi=residuals(hodg.fit,type="deviance")
par(mfrow=c(1,1))
plot(s,d.resi)

Figure 7.5. Deviance plot for Auto-Allo BMT.
§7.6 Checking for influential points

• Influence measure

The influence of individual $j$ is measured by $b - b_{(j)}$, 

$b$: MLE of $\beta$ using all data,  

$b_{(j)}$: MLE of $\beta$ using data with the $j$th observation deleted.

The observations with larger absolute values of the differences have larger influence, those with difference close to zero have little influence.

• Computation

The approximate $b - b_{(j)}$ can be obtained by the R function `residuals`. 

R codes:

```r
coxph.fit=coxph(...) dfb.resid=residuals(coxph.fit,type="dfbeta")

n=dim(dfb.resid)[1] m=dim(dfb.resid)[2] observation.number=c(1:n) for (j in 1:m) {
  plot(observation.number,dfb.resid[,j]
}
```

Example: BMT for Hodgkin and non-Hodgkin’s lymphoma (cont.)

```r
dfbeta.resi=residuals(hodg.fit,type="dfbeta")

n=dim(dfbeta.resi)[1] m=dim(dfbeta.resi)[2] observation.number=c(1:n)

par(mfrow=c(3,2)) for (j in 1:m) {
  plot(observation.number,dfbeta.resi[,j])
}
```
Figure 7.6 Plot for checking influential points for Hodgkin and non-Hodgkin’s lymphoma data