1 Illustrations and Applications

This document reproduces the data analyses presented in Hothorn et al. (2006). For a description of the theory behind applications shown here we refer to the original manuscript. The results differ slightly due to technical changes or bug-fixes in mboost that have been implemented after the paper was printed.

1.1 Acute myeloid leukemia

Data preprocessing Compute IPC weights, define risk score and set up learning sample:

R> ### compute IPC weights
R> AMLw <- IPCweights(Surv(clinical=time, clinical=event))
### risk score

```r
risk <- rep(0, nrow(clinical))
rlev <- levels(clinical[, "Cytogenetic.group"])
risk[clinical[, "Cytogenetic.group"] %in% rlev[c(7, 8, 4)]] <- "low"
risk[clinical[, "Cytogenetic.group"] %in% rlev[c(5, 9)]] <- "intermediate"
risk[clinical[, "Cytogenetic.group"] %in% rlev[-c(4, 5, 7, 8, 9)]] <- "high"
risk <- as.factor(risk)
```

### set-up learning sample

```r
                  risk = risk,
                  iexpressions[, colnames(iexpressions) %in% selgenes[["Clone.ID"]]])
cc <- complete.cases(AMLlearn)
AMLlearn <- AMLlearn[AMLw > 0 & cc,]
AMLw <- AMLw[AMLw > 0 & cc]
```

Model fitting

- Fit random forest for censored data
  ```r
  ### controls for tree growing
  ctrl <- ctree_control(testtype = "Teststatistic",
                        teststat = "maximum", mincriterion = .1, minsplit = 5)
  ### was: cforest_control(mincriterion = 0.1, mtry = 5, minsplit = 5, ntree = 250)
  ### fit random forest for censored data (warnings are OK here)
  AMLrf <- cforest(log(time) ~ ., data = AMLlearn, control = ctrl,
                   weights = AMLw, mtry = 5, ntree = 250,
                   perturb = list(replace = TRUE, fraction = 0.632))
  and $L_2$ Boosting for censored data
  ```

  ```r
  AMLl2b <- glmboost(I(log(time)) ~ ., data = AMLlearn, weights = AMLw,
                     control = boost_control(mstop = 5000))
  ```

Compute fitted values

```r
### restrict number of boosting iterations and inspect selected variables
AML2b <- AML2b[mstop(aic)]
cAML <- coef(AML2b)
cAML[abs(cAML) > 0]
```

<table>
<thead>
<tr>
<th>(Intercept)</th>
<th>Age</th>
<th>WBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.56429</td>
<td>0.00598</td>
<td>-0.00562</td>
</tr>
<tr>
<td>MLL.PTDyes</td>
<td>Tx.Group.AUTO</td>
<td>Tx.Group.Ind</td>
</tr>
<tr>
<td>-0.31539</td>
<td>0.45430</td>
<td>-2.12161</td>
</tr>
<tr>
<td>'IMAGE:145643'</td>
<td>'IMAGE:345601'</td>
<td>'IMAGE:377560'</td>
</tr>
<tr>
<td>0.10626</td>
<td>0.00430</td>
<td>0.02757</td>
</tr>
<tr>
<td>'IMAGE:2043415'</td>
<td>'IMAGE:1584563'</td>
<td>'IMAGE:347035'</td>
</tr>
<tr>
<td>0.05509</td>
<td>-0.00259</td>
<td>-0.00848</td>
</tr>
<tr>
<td>'IMAGE:262695'</td>
<td>'IMAGE:26418'</td>
<td>'IMAGE:950479'</td>
</tr>
<tr>
<td>0.02696</td>
<td>0.00802</td>
<td>0.03717</td>
</tr>
</tbody>
</table>
R> ### AIC criterion
R> plot(aic <- AIC(AML12b))

Figure 1: AIC criterion for AML data.
R> ### fitted values
R> AMLprf <- predict(AMLrf, newdata = AMLlearn)
R> AMLpb <- predict(AMLl2b, newdata = AMLlearn)

1.2 Node-positive breast cancer

Data preprocessing  Compute IPC weights and set up learning sample:

R> ### attach data
R> data("GBSG2", package = "TH.data")
R> ### IPC weights
R> GBSG2w <- IPCweights(Surv(GBSG2$time, GBSG2$cens))
R> ### set-up learning sample
R> GBSG2learn <- cbind(GBSG2[,-which(names(GBSG2) %in% c("time", "cens")],
          ltime = log(GBSG2$time))
R> n <- nrow(GBSG2learn)

Model fitting

R> ### linear model
R> LMmod <- lm(ltime ~ . , data = GBSG2learn, weights = GBSG2w)
R> LMerisk <- sum((GBSG2learn$ltime - predict(LMmod))^2*GBSG2w) / n
R> ### regression tree
R> pos <- GBSG2w > 0
Figure 2: AML data: Reproduction of Figure 1.
R> TRmod <- rpart(ltime ~ ., data = GBSG2learn, weights = GBSG2w,
subset = pos)
R> TRerisk <- sum((GBSG2learn$ltime[pos] - predict(TRmod))^2*GBSG2w[pos]) / n
R> ### tree controls
R> ctrl <- ctree_control(testtype = "Teststatistic",
  teststat = "maximum", mincriterion = qnorm(.95),
  minsplit = 5)
R> ### was: cforest_control(mincriterion = qnorm(.95), mtry = 5,
R> ### minsplit = 5, ntree = 100)
R>
R> ### fit random forest for censored data (warnings are OK here)
R> RFmod <- cforest(ltime ~ ., data = GBSG2learn, weights = GBSG2w,
  control = ctrl, mtry = 5, ntree = 100,
  perturb = list(replace = TRUE,
  fraction = 0.632 * sum(GBSG2w > 0)))
R> ### fit L2 boosting for censored data
R> L2Bmod <- glmboost(ltime ~ ., data = GBSG2learn, weights = GBSG2w,
  control = boost_control(mstop = 250))
R> ### with Huber loss function
R> L2BHubermod <- glmboost(ltime ~ ., data = GBSG2learn, weights = GBSG2w,
  family = Huber(d = log(2)))

Compute fitted values:

R> GBSG2Hp <- predict(L2BHubermod, newdata = GBSG2learn)
R> L2Berisk <- sum((GBSG2learn$ltime - predict(L2Bmod, newdata = GBSG2learn))^2*GBSG2w) / n
R> RFerisk <- sum((GBSG2learn$ltime - predict(RFmod, newdata = GBSG2learn))^2*GBSG2w) / n
R> plot(aic <- AIC(L2Bmod))

Figure 3: AIC criterion for GBSG2 data.
Figure 4: GBSG-2 data: Reproduction of Figure 3.
Figure 5: GBSG-2 data: Reproduction of Figure 5.
Figure 6: GBSG-2 data: Reproduction of Figure 6.
Figure 7: GBSG-2 data: Reproduction of Figure 7.
References