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))
R> ### risk score
R> risk <- rep(0, nrow(clinical))
R> rlev <- levels(clinical[, "Cytogenetic.group"])
R> risk[clinical[, "Cytogenetic.group"] %in% rlev[c(7,8,4)] <- "low"
R> risk[clinical[, "Cytogenetic.group"] %in% rlev[c(5, 9)] <- "intermediate"
R> risk[clinical[, "Cytogenetic.group"] %in% rlev[-c(4,5, 7,8,9)] <- "high"
R> risk <- as.factor(risk)
R> ### set-up learning sample
               risk = risk,
               iexpressions[, colnames(iexpressions) %in% selgenes["Clone.ID"]])
R> cc <- complete.cases(AMLlearn)
R> AMLlearn <- AMLlearn[AMLw > 0 & cc,
R> AMLw <- AMLw[AMLw > 0 & cc]

Model fitting  Fit random forest for censored data
R> ### controls for tree growing
R> ctrl <- cforest_control(mincriterion = 0.1, mtry = 5, minsplit = 5, ntree = 250)
R> ### fit random forest for censored data (warnings are OK here)
R> AMLrf <- cforest(I(log(time)) ~ ., data = AMLlearn, control = ctrl,
                        weights = AMLw)

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
R> AMLl2b <- AMLl2b[mstop(aic)]
R> cAML <- coef(AMLl2b)
R> cAML[abs(cAML) > 0]

(Intercept)    Age       WBC
 0.5642932  0.0059785 -0.0056200
 MLL.PTDyes  Tx.Group.AUTO  Tx.Group.Ind
 0.3153912  0.4542954  2.1216104
 'IMAGE:145643' 'IMAGE:345601'  'IMAGE:377560'
 0.1062577  0.0043043  0.0275653
 'IMAGE:2043415' 'IMAGE:1584563' 'IMAGE:347035'
 0.0550938  0.0025929  0.0084766
 'IMAGE:262695' 'IMAGE:26418' 'IMAGE:950479'
 0.0269555  0.0080214  0.0371741
 'IMAGE:1534700' 'IMAGE:1472689' 'IMAGE:1526826'
 0.0283645  0.0225640 -0.0278373
 'IMAGE:786302' 'IMAGE:243614' 'IMAGE:417884'
 0.0449326 -0.0566722 -0.0249869
 'IMAGE:1592006' 'IMAGE:884333' 'IMAGE:133273'
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)

### fitted values

**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$ltime - predict(LMmod))^2*GBSG2w) / n
R> ### regression tree
R> pos <- GBSG2w > 0
R> TRmod <- rpart(ltime ~ . , data = GBSG2learn, weights = GBSG2w,
>                 subset = pos)
R> TReisk <- sum((GBSG2learn$ltime[pos] - predict(TRmod))^2*GBSG2w[pos]) / n
R> ### tree controls
Figure 2: AML data: Reproduction of Figure 1.
R> ctrl <- cforest_control(mincriterion = qnorm(0.95), mtry = 5,
  minsplit = 5, ntree = 100)
R> ### fit random forest for censored data (warnings are OK here)
R> RFmod <- cforest(ltime ~ ., data = GBSG2learn, weights = GBSG2w,
  control = ctrl)
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