mboost Illustrations

Torsten Hothorn[1] and Peter Bühlmann[2]

[1] Institut für Medizininformatik, Biometrie und Epidemiologie
Friedrich-Alexander-Universität Erlangen-Nürnberg
Waldstraße 6, D-91054 Erlangen, Germany
`Torsten.Hothorn@R-project.org`

[1] Seminar für Statistik
ETH Zürich, CH-8092 Zürich, Switzerland
`buhlmann@stat.math.ethz.ch`

1 Illustrations

This document reproduces the data analyses presented in Bühlmann and Hothorn (2006). For a description of the theory behind applications shown here we refer to the original manuscript. Note: The *Breast Cancer Subtypes* example is missing from this document because we cannot assume package **Biobase** to be installed.

Illustration: Prediction of Total Body Fat Garcia et al. (2005) report on the development of predictive regression equations for body fat content by means of $p = 9$ common anthropometric measurements which were obtained for $n = 71$ healthy German women. In addition, the women's body composition was measured by Dual Energy X-Ray Absorptiometry (DXA). This reference method is very accurate in measuring body fat but finds little applicability in practical environments, mainly because of high costs and the methodological efforts needed. Therefore, a simple regression equation for predicting DXA measurements of body fat is of special interest for the practitioner. Backward-elimination was applied to select important variables from the available anthropometrical measurements and Garcia et al. (2005) report a final linear model utilizing hip circumference, knee breadth and a compound covariate which is defined as the sum of log chin skinfold, log triceps skinfold and log subscapular skinfold:

```
R> bf_lm <- lm(DEXfat ~ hipcirc + kneebreadth + anthro3a,
         data = bodyfat)
R> coef(bf_lm)
```

```
(Intercept)     hipcirc kneebreadth    anthro3a
  -75.23478     0.51153     1.90199     8.90964
```

Since a simple and easy to communicate regression formula, such as a linear combination of only a few covariates, is of special interest in this application, we employ the `glmboost` function from package **mboost** to fit a linear regression model by means of L_2Boosting with componentwise linear least squares. We first center the covariates and specify a formula describing the model we want to fit:

```
R> indep <- names(bodyfat)[names(bodyfat) != "DEXfat"]
R> cbodyfat <- bodyfat
R> cbodyfat[indep] <- lapply(cbodyfat[indep], function(x) x -
         mean(x))
R> bffm <- DEXfat ~ age + waistcirc + hipcirc + elbowbreadth +
         kneebreadth + anthro3a + anthro3b + anthro3c +
         anthro4
```

By default, the function `glmboost` fits a linear model (with initial $m_{\text{stop}} = 100$ and shrinkage parameter $\nu = 0.1$) by minimizing squared error (argument `family = GaussReg()` is the default):

```
R> bf_glm <- glmboost(bffm, data = cbodyfat)
```

Note that, by default, the mean of the response variable is used as an offset in the first step of the boosting algorithm. As mentioned above, the special form of the base learner, i.e., componentwise linear least squares, allows for a reformulation of the boosting fit in terms of a linear combination of the covariates which can be assessed via

```
R> coef(bf_glm)
```

```
 (Intercept)          age    waistcirc      hipcirc
    0.000000     0.013602     0.189716     0.351626
elbowbreadth  kneebreadth     anthro3a     anthro3b
   -0.384140     1.736589     3.326860     3.656524
    anthro3c      anthro4
    0.595363     0.000000
```

We notice that most covariates have been used for fitting and thus no extensive variable selection was performed in the above model. Thus, we need to investigate how many boosting iterations are appropriate. Resampling methods such as cross-validation or the bootstrap can be used to study the empirical risk for a varying number of boosting iterations. The out-of-bootstrap mean squared error for 100 bootstrap samples is depicted in the upper part of Figure 1. The plot leads to the impression that approximately $m_{\text{stop}} = 44$ would be a sufficient number of boosting iterations. In Section **??**, a corrected version of the Akaike information criterion (AIC) is proposed for determining the optimal number of boosting iterations. This criterion attains its minimum for

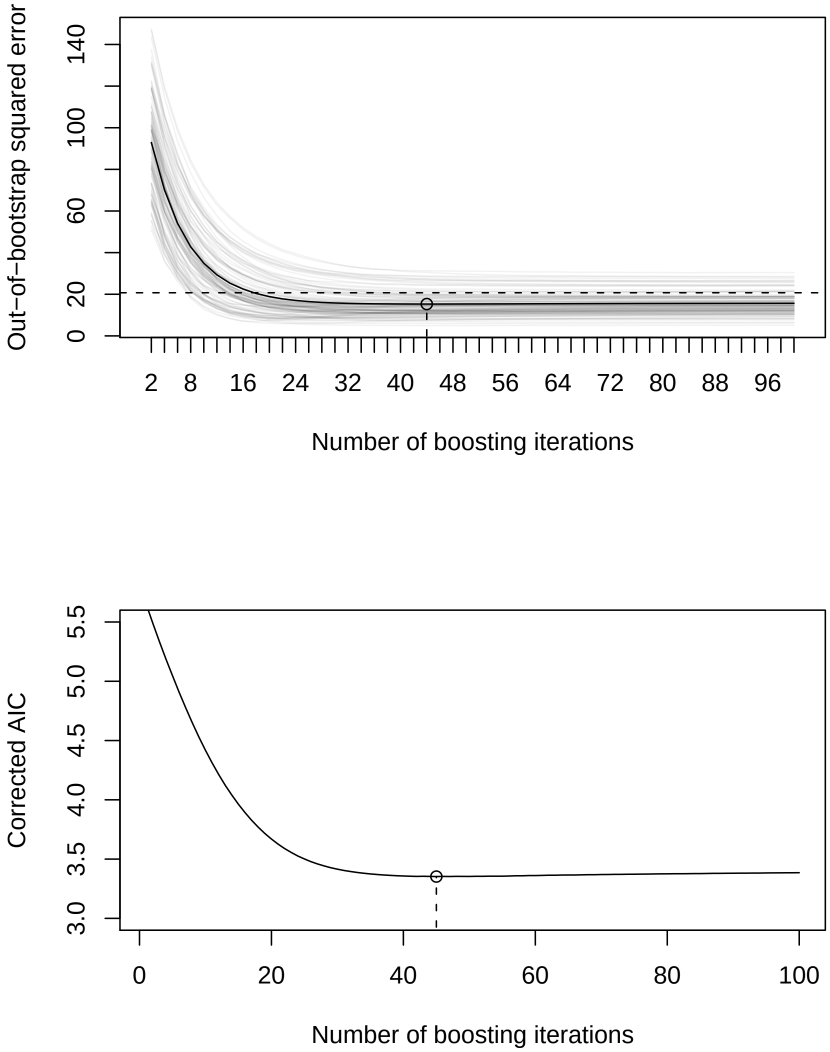

Figure 1: `bodyfat` data: Out-of-bootstrap squared error for varying number of boosting iterations m_{stop} (top). The dashed horizontal line depicts the out-of-bootstrap error of the linear model for the pre-selected variables `hipcirc`, `kneebreadth` and `anthro3a` fitted via ordinary least squares. The lower part show the corrected AIC criterion.

```
R> mstop(aic <- AIC(bf_glm))
```

```
[1] 45
```

boosting iterations, see the bottom part of Figure 1 in addition.

The coefficients of the boosted linear model with $m_{\text{stop}} = 45$ boosting iterations are

```
R> coef(bf_glm[mstop(aic)])
```

```
 (Intercept)          age    waistcirc      hipcirc
   0.0000000    0.0023271    0.1893046    0.3488781
elbowbreadth  kneebreadth     anthro3a     anthro3b
   0.0000000    1.5217686    3.3268603    3.6051548
    anthro3c      anthro4
   0.5043133    0.0000000
```

and thus only 7 covariates have been selected for the final model (intercept equal to zero occurs here for mean centered response and predictors and hence, $n^{-1} \sum_{i=1}^{n} Y_i = 30.783$ is the intercept in the uncentered model). Note that the variables hipcirc, kneebreadth and anthro3a, which we have used for fitting a simple linear model at the beginning of this paragraph, have been selected by the boosting algorithm as well.

Illustration: Prediction of Total Body Fat (cont.) Being more flexible than the linear model which we fitted to the bodyfat data in Section **??**, we estimate an additive model using the gamboost function from **mboost** (first with pre-specified $m_{\text{stop}} = 100$ boosting iterations, $\nu = 0.1$ and squared error loss):

```
R> bf_gam <- gamboost(bffm, data = cbodyfat)
```

The degrees of freedom for the smoothing splines, which are utilized as base learners here, can be defined by the dfbase argument, defaulting to 4.

We can estimate the number of boosting iterations m_{stop} using the corrected AIC criterion described in Section **??** (see Figure 2) via

```
R> mstop(aic <- AIC(bf_gam))
```

```
[1] 46
```

```
R> bf_gam <- bf_gam[mstop(aic)]
```

Similar to the linear regression model, the partial contributions of the covariates can be extracted from the boosting fit. For the most important variables, the partial fits are given in Figure 3 showing some slight non-linearity for hipcirc and kneebreadth.

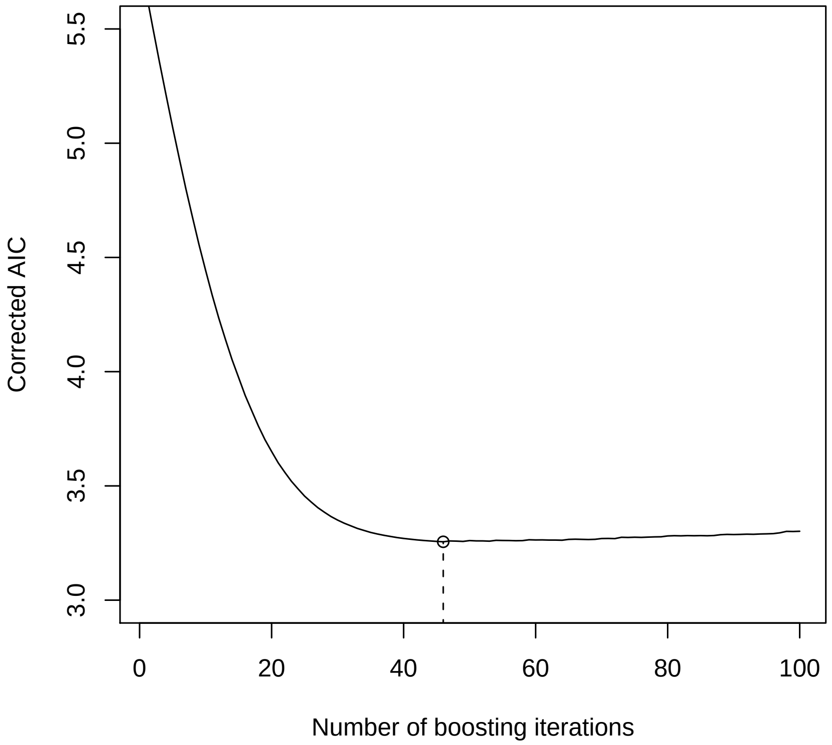

Figure 2: `bodyfat` data: Corrected AIC as a function of the number of boosting iterations m_{stop} for fitting an additive model via `gamboost`.

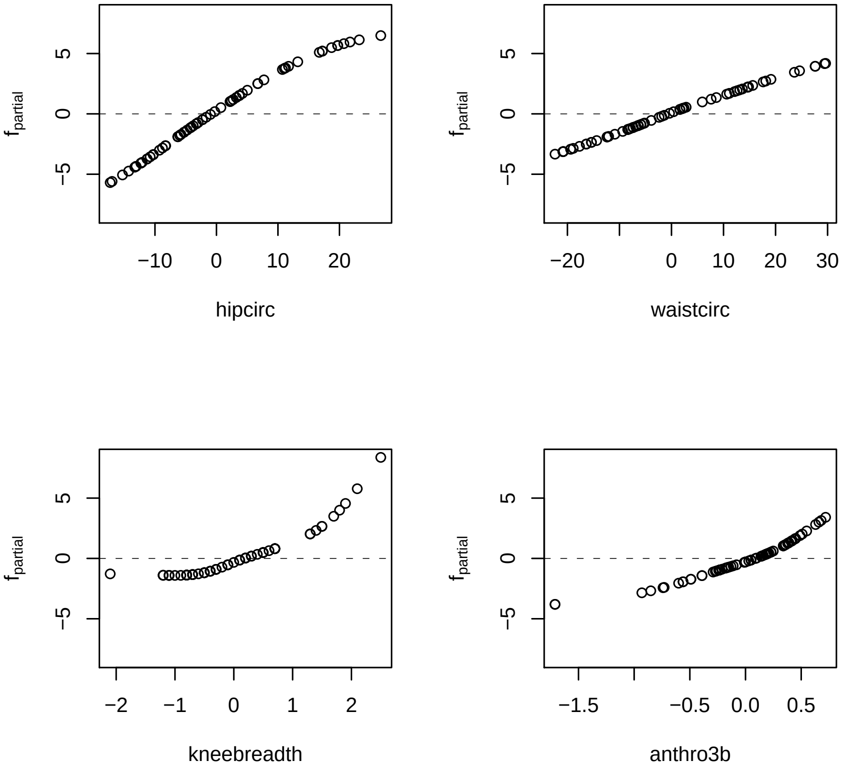

Figure 3: `bodyfat` data: Partial contributions of four covariates in an additive model.

Illustration: Prediction of Total Body Fat (cont.) Such transformation defined in terms of fractional polynomials for estimating a linear model can be used with the `glmboost` function (an implementation of the original fractional polynomials approach is available in package **mfp**, see Ambler and Benner, 2005; Sauerbrei et al., 2006), where the model formula performs the computations of all transformations by means of the `FP` (fractional polynomials) function. First, we scale each covariate to the interval $[1, 2]$ and then fit the complex linear model by using the `glmboost` function with initial $m_{\text{stop}} = 3000$ boosting iterations:

```
R> tbodyfat <- bodyfat
R> tbodyfat[indep] <- lapply(bodyfat[indep], function(x) {
         x <- x - min(x)
         x/max(x) + 1
     })
R> fpfm <- as.formula(paste("DEXfat ~ ", paste("FP(",
         indep, ")", collapse = "+")))
R> fpfm
```

```
DEXfat ~ FP(age) + FP(waistcirc) + FP(hipcirc) + FP(elbowbreadth) +
    FP(kneebreadth) + FP(anthro3a) + FP(anthro3b) + FP(anthro3c) +
    FP(anthro4)
```

```
R> bf_fp <- glmboost(fpfm, data = tbodyfat, control = boost_control(mstop = 3000))
R> mstop(aic <- AIC(bf_fp))
```

```
[1] 2480
```

The corrected AIC criterion (see Section **??**) suggests to stop after $m_{\text{stop}} =$ 2480 boosting iterations and the final model selects 21 (transformed) predictor variables. Again, the partial contributions of each of the 9 original covariates can be computed easily and are shown (for the same variables as in Figure 3) in Figure 4. Note that the depicted functional relationship derived from the multivariate fractional polynomial model (Figure 4) is qualitatively the same as the one derived from the additive model (Figure 2).

Illustration: Wisconsin Prognostic Breast Cancer Prediction models for recurrence events in breast cancer patients based on covariates which have been computed from a digitized image of a fine needle aspirate of breast tissue (those measurements describe characteristics of the cell nuclei present in the image) have been studied by Street et al. (1995) (the data is part of the UCI repository Blake and Merz, 1998).

We first analyse this data as a binary prediction problem (recurrence vs. non-recurrence) and later in Section **??** by means of survival models. Again, we are faced with lots of potential covariates ($p = 32$) for a limited number of observations without missing values ($n = 194$) and variable selection is an issue. We can choose a classical logistic regression model via AIC in a stepwise algorithm as follows (after centering the covariates)

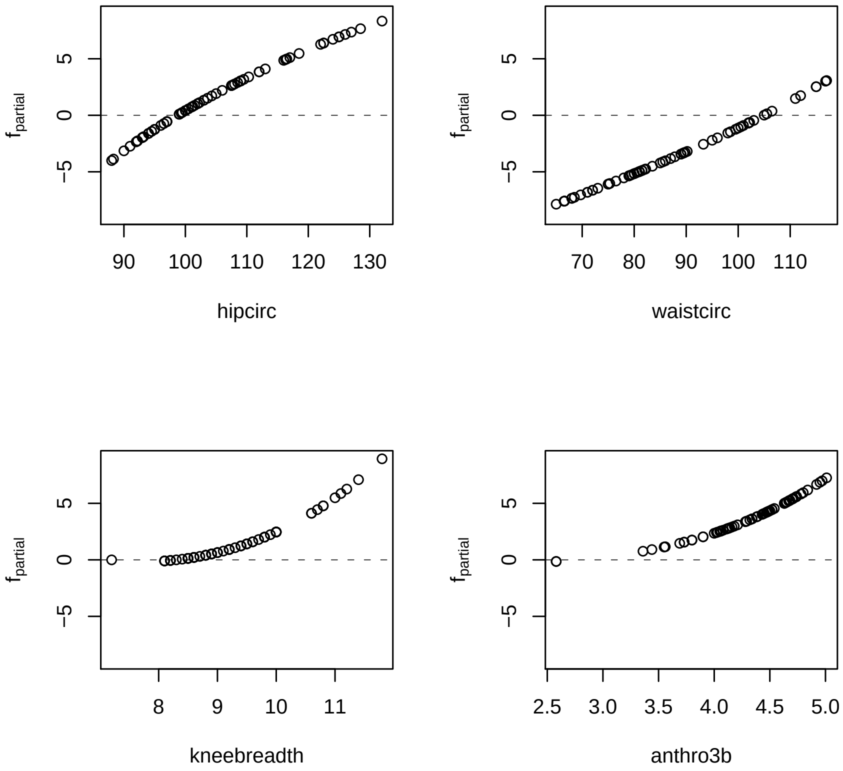

Figure 4: `bodyfat` data: Partial fits for a linear model including fractional polynomials.

```
R> wpbc2 <- wpbc[complete.cases(wpbc), colnames(wpbc) !=
         "time"]
R> indep <- names(wpbc2)[names(wpbc2) != "status"]
R> wpbc2[indep] <- lapply(wpbc2[indep], function(x) x -
         mean(x))
R> wpbc_step <- step(glm(status ~ ., data = wpbc2,
         family = binomial()), trace = 0)
```

The final model consists of 16 parameters with

```
R> logLik(wpbc_step)
```

```
'log Lik.' -80.13 (df=16)
```

```
R> AIC(wpbc_step)
```

```
[1] 192.26
```

and we want to compare this model to a logistic regression model fitted via gradient boosting. We simply select the `Binomial` family (with default offset of $1/2 \log(p/(1-p))$, where p is the proportion of recurrences) and start with initial $m_{\text{stop}} = 500$ boosting iterations

```
R> wpbc_glm <- glmboost(status ~ ., data = wpbc2,
         family = Binomial(), control = boost_control(mstop = 500))
```

The negative binomial log-likelihood is

```
R> logLik(wpbc_glm)
```

```
[1] -89.964
```

and the classical AIC criterion suggests to stop after

```
R> aic <- AIC(wpbc_glm, "classical")
R> aic
```

```
[1] 198.44
Optimal number of boosting iterations: 260
Degrees of freedom (for mstop = 260): 7.032
```

boosting iterations. We now restrict the number of boosting iterations to $m_{\text{stop}} = 260$ via

```
R> wpbc_glm <- wpbc_glm[mstop(aic)]
R> logLik(wpbc_glm)
```

```
[1] -92.189
```

```
R> coef(wpbc_glm)[abs(coef(wpbc_glm)) > 0]
```

```
     (Intercept)     mean_texture    mean_symmetry
     -3.0110e-02      -2.4215e-02      -3.3878e+00
 mean_fractaldim       SE_texture     SE_perimeter
     -2.0321e+01      -2.6603e-02       4.0908e-02
  SE_compactness     SE_concavity SE_concavepoints
      7.0280e+00      -4.6303e+00      -1.5737e+01
     SE_symmetry     worst_radius  worst_perimeter
      2.8601e+00       1.7777e-02       1.2639e-03
      worst_area worst_smoothness            tsize
      1.5854e-04       8.8372e+00       3.1014e-02
          pnodes
      2.5981e-02
```

(because of using the offset-value $\hat{f}^{[0]}$, we have to add the value $\hat{f}^{[0]}$ to the reported intercept estimate above for the logistic regression model) and we then can extract the fitted conditional probabilities

```
R> f <- fitted(wpbc_glm)
R> p <- exp(f)/(exp(f) + exp(-f))
```

which are depicted by a conditional density plot in Figure 5.

A generalized additive model adds more flexibility to the regression function but is still interpretable. We fit a logistic additive model to the wpbc data as follows:

```
R> wpbc_gam <- gamboost(status ~ ., data = wpbc2,
         family = Binomial())
R> mopt <- mstop(aic <- AIC(wpbc_gam, "classical"))
R> aic
```

```
[1] 196.33
Optimal number of boosting iterations: 84
Degrees of freedom (for mstop = 84): 13.754
```

This model selected 16 out of 32 covariates. The partial contributions of the four most important variables are depicted in Figure 6 indicating a remarkable degree of non-linearity.

Illustration: Wisconsin Prognostic Breast Cancer (cont.) Instead of the binary response variable describing the recurrence status we make use of the additionally available time information for modeling the time to recurrence, i.e., all observations with non-recurrence are censored. First, we calculate IPC weights and center the covariates

```
R> iw <- IPCweights(Surv(wpbc$time, wpbc$status ==
         "R"))
R> wpbc3 <- wpbc[, colnames(wpbc) != "status"]
R> indep <- names(wpbc3)[names(wpbc3) != "time"]
```

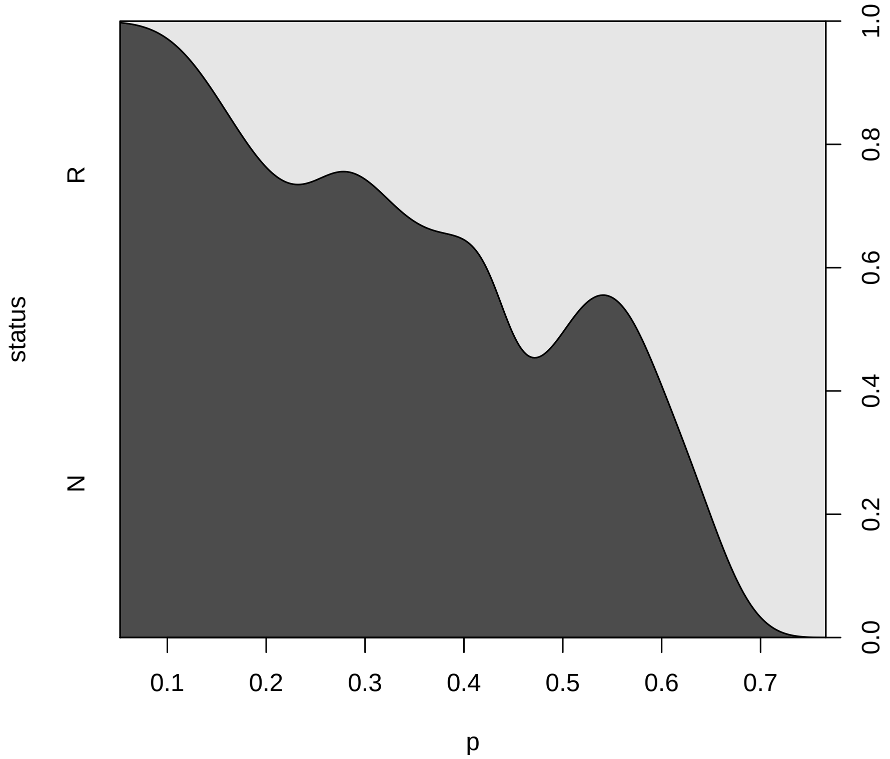

Figure 5: `wpbc` data: Conditional density plot of the fitted probabilities of recurrence / non-recurrence.

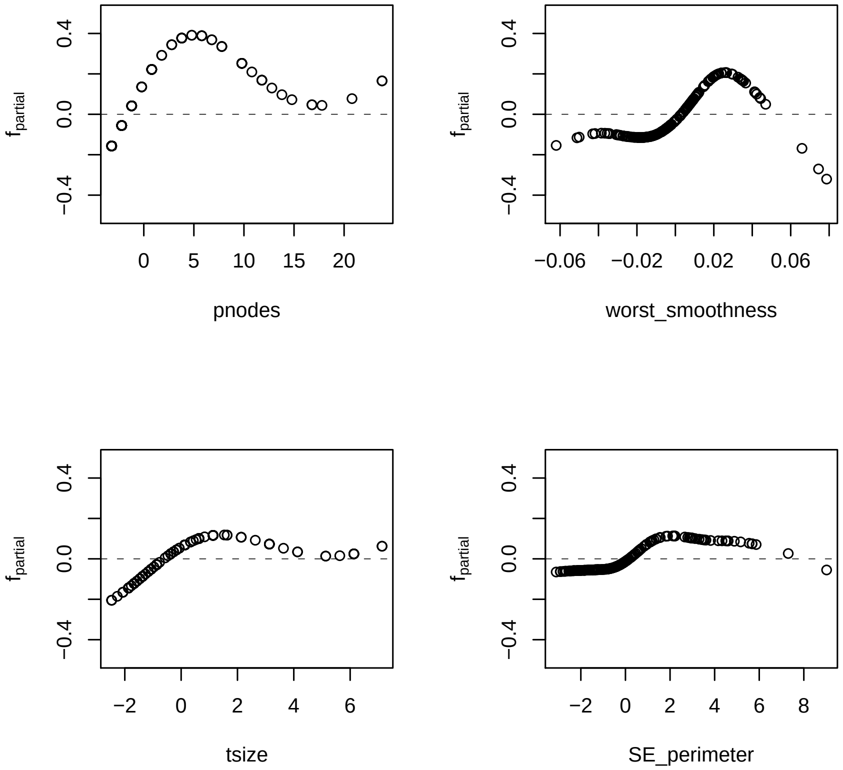

Figure 6: `wpbc` data: Partial contributions of four selected covariates in an additive logistic model.

```
R> wpbc3[indep] <- lapply(wpbc3[indep], function(x) x -
         mean(x, na.rm = TRUE))
```

and fit a weighted linear model by boosting with componentwise linear weighted least squares as base procedure:

```
R> wpbc_surv <- glmboost(log(time) ~ ., data = wpbc3,
         control = boost_control(mstop = 500), weights = iw)
R> mstop(aic <- AIC(wpbc_surv))

[1] 122

R> wpbc_surv <- wpbc_surv[mstop(aic)]
```

The following variables have been selected for fitting

```
R> names(coef(wpbc_surv)[abs(coef(wpbc_surv)) > 0])

 [1] "mean_radius"         "mean_texture"
 [3] "mean_perimeter"      "mean_smoothness"
 [5] "mean_symmetry"       "SE_texture"
 [7] "SE_smoothness"       "SE_concavepoints"
 [9] "SE_symmetry"         "worst_concavepoints"
```

and the fitted values are depicted in Figure 7, showing a reasonable model fit.

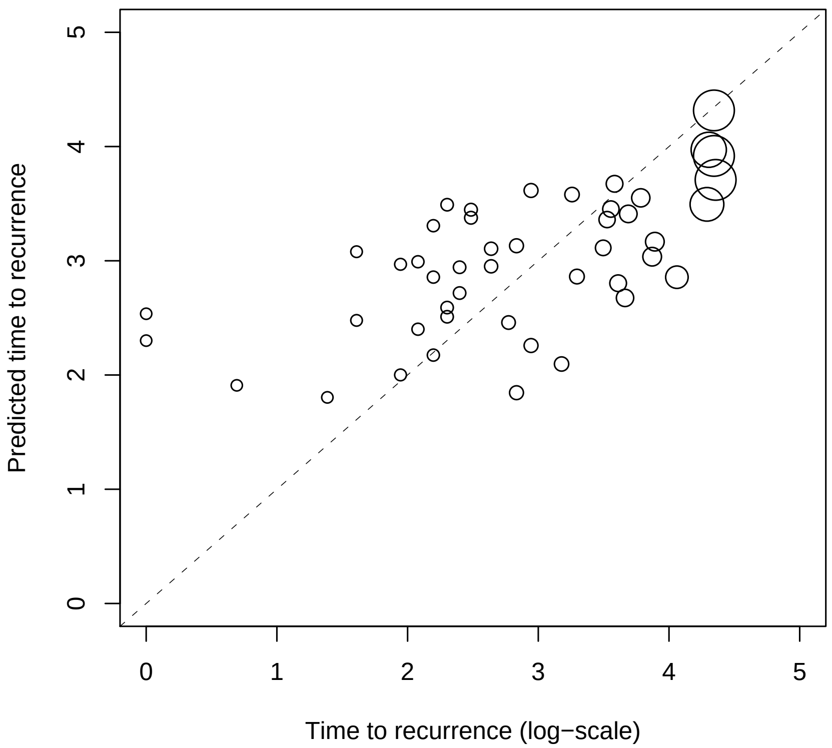

Figure 7: `wpbc` data: Fitted values of a weighted linear model taking both time to recurrence and censoring information into account. The radius of the circles is proportional to the IPC weight of the corresponding observation, censored observations with IPC weight zero are not plotted.

References

Gareth Ambler and Axel Benner. *mfp: Multivariable Fractional Polynomials*, 2005. URL http://CRAN.R-project.org. R package version 1.3.2.

C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998. URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

P. Bühlmann and T. Hothorn. Boosting: A statistical perspective. 2006. submitted manuscript.

A. L. Garcia, K. Wagner, T. Hothorn, C. Koebnick, H. J. Zunft, and U. Trippo. Improved prediction of body fat by measuring skinfold thickness, circumferences, and bone breadths. *Obesity Research*, 13(3):626–634, 2005.

W. Sauerbrei, C. Meier-Hirmer, A. Benner, and P. Royston. Multivariable regression model building by using fractional polynomials: Description of SAS, STATA and R programs. *Computational Statistics & Data Analysis*, 2006. in press.

W. N. Street, O. L. Mangasarian, , and W. H. Wolberg. An inductive learning approach to prognostic prediction. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 522–530, San Francisco, CA, 1995. Morgan Kaufmann Publishers Inc.