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

[2] 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 (2008). 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
```

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. 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(DEXfat ~ ., data = bodyfat,
         control = boost_control(center = TRUE))
```

Note that, by default, the mean of the response variable is used as an offset in the first step of the boosting algorithm. We center the covariates prior to model fitting in addition. 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
attr(,"offset")
[1] 30.783
```

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 estimate the out-of-sample error 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

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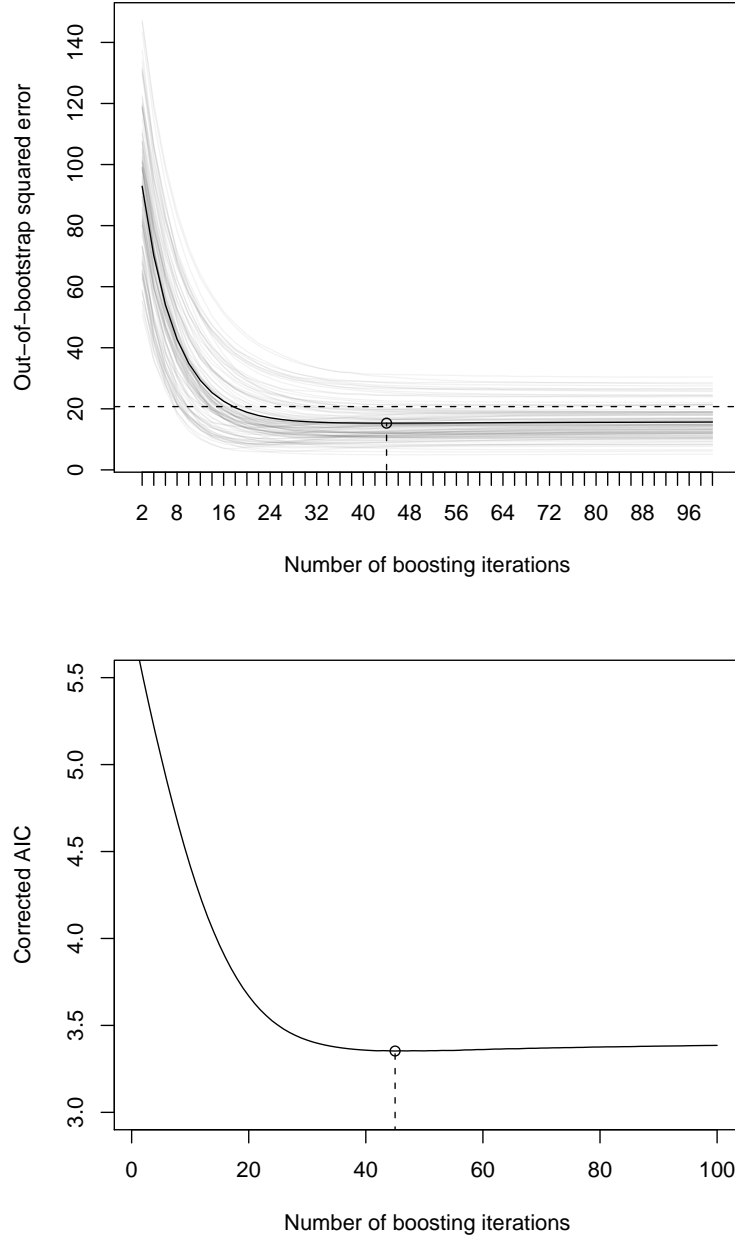
Figure 1: `bodyfat` data: Out-of-bootstrap squared error for varying number of boosting iterations m_{stop} (top). The dashed horizontal line depicts the average out-of-bootstrap error of the linear model for the pre-selected variables `hipcirc`, `kneebreadth` and `anthro3a` fitted via ordinary least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

3

```
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
attr(,"offset")
[1] 30.783
```

and thus 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 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(DEXfat ~ ., data = bodyfat)
```

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 **??** via

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

```
[1] 46
```

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 2 showing some slight non-linearity, mainly for `kneebreadth`.

Illustration: Prediction of total body fat (cont.) Such transformations and estimation of a corresponding linear model can be done with the `glmboost` function, where the model formula performs the computations of all transformations by means of the `bs` (B-spline basis) function from package **splines**. First, we set up a formula transforming each covariate

```
R> bsfm
```

```
DEXfat ~ bs(age) + bs(waistcirc) + bs(hipcirc) + bs(elbowbreadth) +
    bs(kneebreadth) + bs(anthro3a) + bs(anthro3b) + bs(anthro3c) +
    bs(anthro4)
```
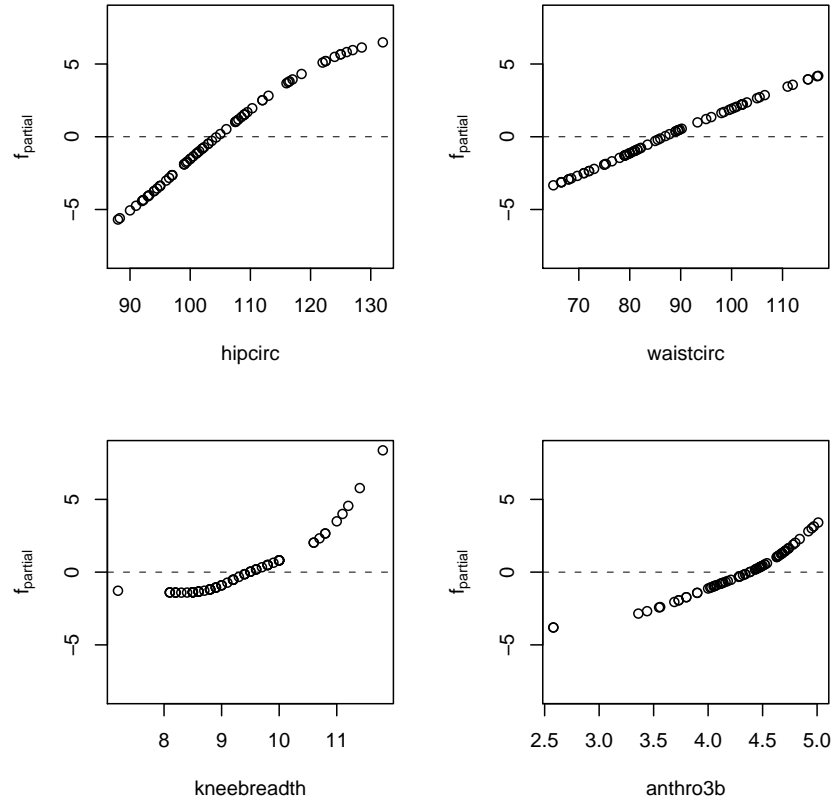
4

Figure 2: `bodyfat` data: Partial contributions of four covariates in an additive model (without centering of estimated functions to mean zero).

and then fit the complex linear model by using the `glmboost` function with initial $m_{\text{stop}} = 5000$ boosting iterations:

```
R> ctrl <- boost_control(mstop = 5000)
R> bf_bs <- glmboost(bsfm, data = bodyfat, control = ctrl)
R> mstop(aic <- AIC(bf_bs))
```

[1] 2891

The corrected AIC criterion (see Section **??**) suggests to stop after $m_{\text{stop}} = 2891$ 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 in Figure 3 (for the same variables as in Figure 2). Note that the depicted functional relationship derived from the

model fitted above (Figure 3) 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 analyze this data as a binary prediction problem (recurrence vs. non-recurrence) and later in Section **??** by means of survival models. We are faced with many 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

```
R> cc <- complete.cases(wpbc)
R> wpbc2 <- wpbc[cc, colnames(wpbc) != "time"]
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 $1/2 \log(\hat{p}/(1 - \hat{p}))$, where $\hat{p}$ is the empirical proportion of recurrences) and we initially use $m_{\mathrm{stop}} = 500$ boosting iterations

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

The classical AIC criterion (-2 log-likelihood + 2 df) suggests to stop after

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

```
[1] 199.54
Optimal number of boosting iterations: 465
Degrees of freedom (for mstop = 465): 9.147
```

boosting iterations. We now restrict the number of boosting iterations to $m_{\mathrm{stop}} = 465$ and then obtain the estimated coefficients via

```
R> wpbc_glm <- wpbc_glm[mstop(aic)]
R> coef(wpbc_glm)[abs(coef(wpbc_glm)) > 0]

     (Intercept)       mean_radius      mean_texture
      -1.2511e-01       -5.8453e-03       -2.4505e-02
  mean_smoothness     mean_symmetry    mean_fractaldim
       2.8513e+00       -3.9307e+00       -2.8253e+01
       SE_texture       SE_perimeter     SE_compactness
      -8.7553e-02        5.4917e-02        1.1463e+01
      SE_concavity   SE_concavepoints       SE_symmetry
      -6.9238e+00       -2.0454e+01        5.2125e+00
     SE_fractaldim       worst_radius    worst_perimeter
       5.2187e+00        1.3468e-02        1.2108e-03
       worst_area   worst_smoothness  worst_compactness
       1.8646e-04        9.9560e+00       -1.9469e-01
            tsize             pnodes
       4.1561e-02        2.4445e-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).

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] 199.76
Optimal number of boosting iterations: 99
Degrees of freedom (for mstop = 99): 14.583
```

This model selected 16 out of 32 covariates. The partial contributions of the four most important variables are depicted in Figure 4 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

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

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

```
R> ctrl <- boost_control(mstop = 500, center = TRUE)
R> wpbc_surv <- glmboost(log(time) ~ ., data = wpbc3,
         control = ctrl, 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 5, showing a reasonable model fit.

Alternatively, a Cox model with linear predictor can be fitted using L_2Boosting by implementing the negative gradient of the partial likelihood (see Ridgeway (1999)) via

```
R> ctrl <- boost_control(center = TRUE)
R> glmboost(Surv(wpbc$time, wpbc$status == "N") ~
         ., data = wpbc, family = CoxPH(), control = ctrl)
```

For more examples, such as fitting an additive Cox model using **mboost**, see (Hothorn and Bühlmann, 2006).
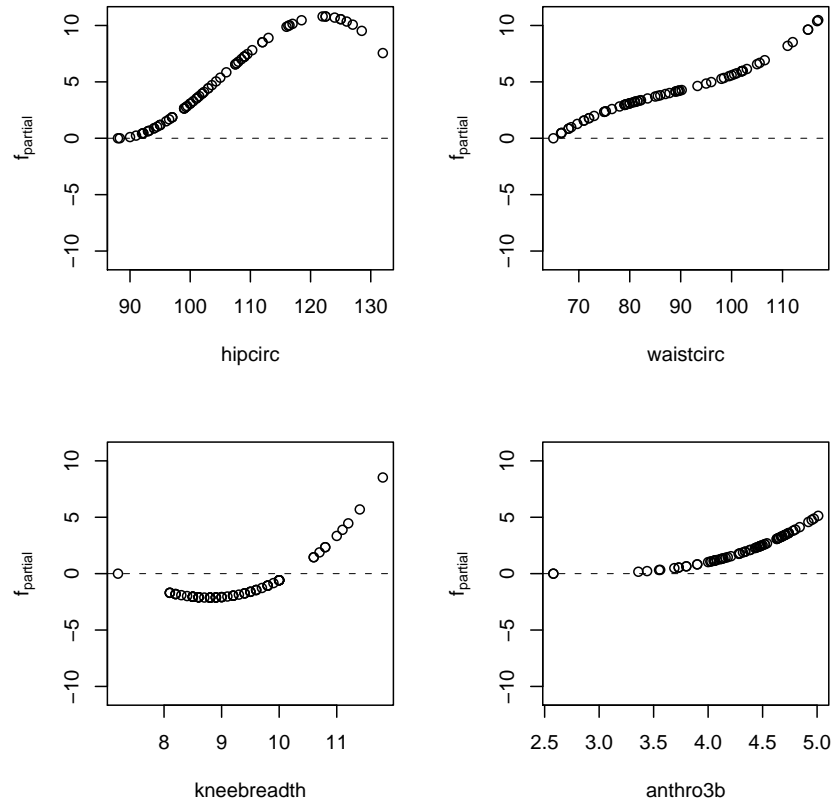
Figure 3: `bodyfat` data: Partial fits for a linear model fitted to transformed covariates using B-splines (without centering of estimated functions to mean zero).
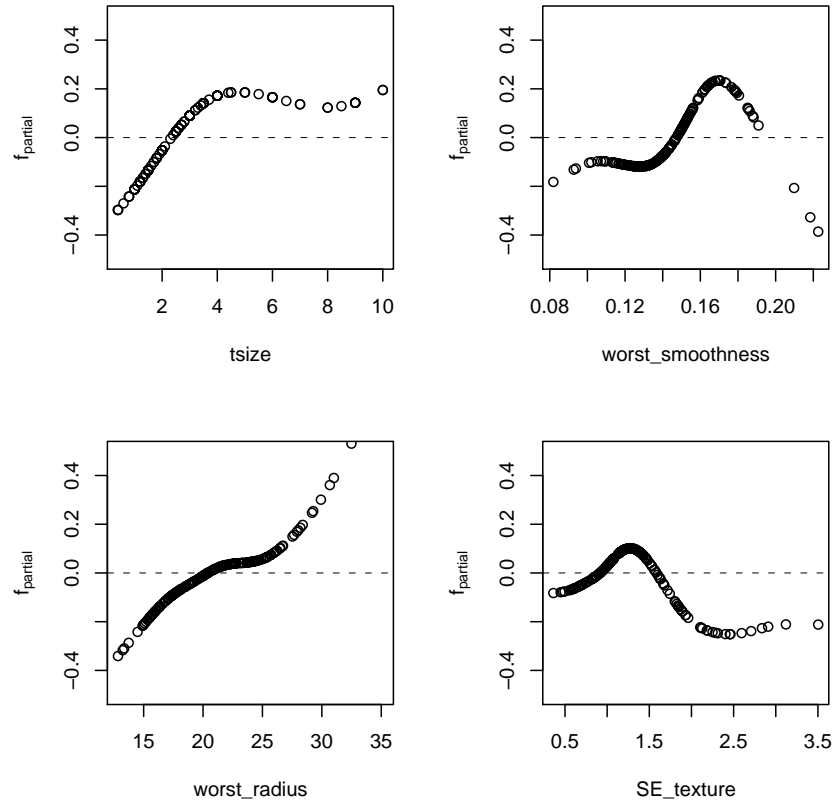
Figure 4: `wpbc` data: Partial contributions of four selected covariates in an additive logistic model (without centering of estimated functions to mean zero).
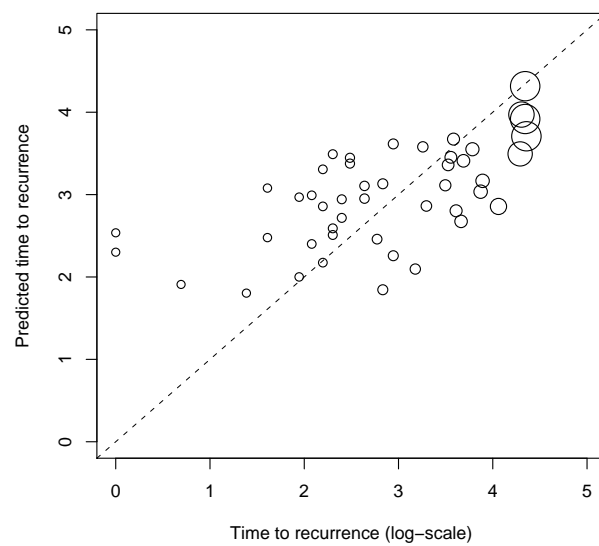
Figure 5: `wpbc` data: Fitted values of an IPC-weighted linear model, taking both time to recurrence and censoring information into account. The radius of the circles is proportional to the IPC weight of the corresponding observation, censored observations with IPC weight zero are not plotted.

References

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 algorithms: regularization, prediction and model fitting. *Statistical Science*, 2008. URL ftp://ftp.stat.math.ethz.ch/Research-Reports/Other-Manuscripts/buhlmann/BuehlmannHothorn_Boosting-rev.pdf. (in press).

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.

T Hothorn and P Bühlmann. Model-based boosting in high dimensions. *Bioinformatics*, 22(22):2828–2829, Nov 2006. doi: doi:10.1093/bioinformatics/btl462.

G. Ridgeway. The state of boosting. *Computing Science and Statistics*, 31: 172–181, 1999.

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.