LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

INSTITUT FÜR STATISTIK

Matthias Schmid, Sergej Potapov,
Annette Pfahlberg & Torsten Hothorn

Estimation and Regularization Techniques for Regression Models with Multidimensional Prediction Functions

Technical Report Number 042, 2008, 2009
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Estimation and Regularization Techniques for Regression Models with Multidimensional Prediction Functions

Matthias Schmid[1] Sergej Potapov[1]
Annette Pfahlberg[1] Torsten Hothorn[2]

[1] Institut für Medizininformatik, Biometrie und Epidemiologie
Friedrich-Alexander-Universität Erlangen-Nürnberg
Waldstraße 6, 91054 Erlangen, Germany
Email: matthias.schmid@imbe.med.uni-erlangen.de

[2] Institut für Statistik
Ludwig-Maximilians-Universität München
Ludwigstraße 33, 80539 München, Germany

Abstract

Boosting is one of the most important methods for fitting regression models and building prediction rules. A notable feature of boosting is that the technique can be modified such that it includes a built-in mechanism for shrinking coefficient estimates and variable selection. This regularization mechanism makes boosting a suitable method for analyzing data characterized by small sample sizes and large numbers of predictors. We extend the existing methodology by developing a boosting method for prediction functions with multiple components. Such multidimensional functions occur in many types of statistical models, for example in count data models and in models involving outcome variables with a mixture distribution. As will be demonstrated, the new algorithm is suitable for both the estimation of the prediction function and regularization of the estimates. In addition, nuisance parameters can be estimated simultaneously with the prediction function.

Keywords: Gradient boosting, multidimensional prediction function, scale parameter estimation, variable selection, count data model.

1 Introduction

A common problem in statistical research is the development of model fitting and prediction techniques for the analysis of high-dimensional data. High-dimensional data sets, which are characterized by relatively small sample sizes and large numbers of variables, arise in many fields of modern research. Most notably, advances in genomic research have led to large sets of gene expression data where sample sizes are considerably smaller than the number of gene expression measurements
(Golub et al. 1999, Dudoit et al. 2002). A consequence of this "$p > n$" situation is that standard techniques for prediction and model fitting (such as maximum likelihood estimation) become infeasible. Moreover, high-dimensional data sets usually involve the problem of separating noise from information, i.e., of selecting a small number of relevant predictors from the full set of variables.

In a regression framework, the problem of analyzing high-dimensional data can be formulated as follows: Consider a data set containing the values of an outcome variable $\boldsymbol{Y}$ and predictor variables $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p$. Although $\boldsymbol{Y}$ will be one-dimensional in most applications, we explicitly allow for multidimensional outcome variables. The objective is to model the relationship between $\boldsymbol{Y}$ and $\boldsymbol{X} := (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p)^\top$, and to obtain an "optimal" prediction of $\boldsymbol{Y}$ given $\boldsymbol{X}$. Usually, this is accomplished by optimizing an objective function $\rho(\boldsymbol{Y}, f, \sigma) \in \mathbb{R}$ over a prediction function f (depending on $\boldsymbol{X}$) and a set of scale parameters (denoted by σ). Linear regression with a continuous outcome variable $\boldsymbol{Y} \in \mathbb{R}$ is a well-known example of this approach: Here, ρ corresponds to the least squares objective function, f is a parametric (linear) function of $\boldsymbol{X}$, and $\sigma \in \mathbb{R}^+$ is the residual variance.

In order to address the issue of analyzing high-dimensional data sets, a variety of regression techniques have been developed over the past years (see, e.g., Hastie et al. 2009). Many of these techniques are characterized by a built-in mechanism for "regularization", which means that shrinkage of coefficient estimates or selection of relevant predictors is carried out simultaneously with the estimation of the model parameters. Both shrinkage and variable selection will typically improve prediction accuracy: In case of shrinkage, coefficient estimates tend to have a slightly increased bias but a decreased variance, while in case of variable selection, overfitting the data is avoided by selecting the most informative predictors only. Note that regularization is not only useful for analyzing high-dimensional data but also tends to improve prediction accuracy in low-dimensional settings where $p \leq n$.

Important examples of recently developed regularization techniques are gradient boosting (which will be considered in this paper) and L_1 penalized estimation. *Gradient boosting* (Breiman 1998, 1999, Friedman et al. 2000, Friedman 2001) is an iterative method for obtaining statistical model estimates via gradient descent techniques. A key feature of gradient boosting is that the procedure can be modified such that variable selection is carried out in each iteration (Bühlmann and Yu 2003, Bühlmann 2006). As a result, the final boosting fit typically depends on only a small subset of predictor variables but can still be

interpreted as the fit of a regression model. The possibility of making estimates interpretable is in fact a major strength of gradient boosting: Although boosting algorithms are tuned to optimize prediction accuracy, uninterpretable "black-box" predictions (which are obtained, e.g., from the random forest method, Breiman 2001) can be avoided. L_1 *penalized estimation* techniques have been developed for regression models with a linear prediction function. Due to the structure of the L_1 penalty, a number of coefficient estimates will typically become zero, so that the procedure implicitly results in a selection of the most informative predictor variables. The most important examples of L_1 penalized techniques are the Lasso and its extensions (Tibshirani 1996, Tibshirani et al. 2005, Zou 2006, Yuan and Lin 2006), SCAD procedures (Fan and Li 2001) and the Elastic Net methodology (being a combination of L_1 and L_2 penalized regression, see Zou and Hastie 2005). By introducing the LARS algorithm for linear prediction functions, Efron et al. (2004) have embedded boosting and L_1 penalized techniques into a more general framework (LARS will, however, not be considered in this paper). Both boosting and L_1 penalized estimation techniques can be applied to a large variety of statistical problems, such as regression, classification and time-to-event analysis (Bühlmann and Hothorn 2007, Park and Hastie 2007). Besides being computationally efficient, the techniques are competitive with methods based on a separation of the variable selection and model fitting processes (see, e.g., Segal 2006).

A limitation of classical boosting and L_1 penalized estimation approaches is that the techniques are designed for statistical problems involving a *one-dimensional* prediction function only. In fact, boosting and L_1 penalized estimation are suitable for fitting many common statistical models, such as linear or logistic regression. However, there is a variety of important statistical problems that cannot be reduced to estimating a one-dimensional prediction function only. This is particularly true when scale parameters or nuisance parameters have to be estimated simultaneously with the prediction function, or when the prediction function itself depends on multiple components. Typical examples of such multidimensional estimation problems are:

(a) *Classification with multiple outcome categories.* Regressing outcome variables with a multinomial distribution on a set of predictor variables is a natural extension of the binary classification problem. In the setting of a multinomial logit model, each of the outcome categories is associated with a separate component of the prediction function. Thus, if there is a total number of K possible outcome categories, a K-dimensional prediction function has to be estimated. Hastie et al. (2009) have addressed this problem by constructing a gradient boosting algorithm for multiclass prediction (see Algorithm 10.4 in Hastie et al. 2009). Apart from multiclass gradient boosting, various other multiclass boosting procedures have been discussed in the literature, see, e.g., Freund and Schapire (1997), Schapire and Singer (1999), Zhu et al. (2005), Li (2006), Sun et al. (2007) and the literature cited there.

(b) *Regression models for count data.* Apart from the classical Poisson model, count data models are typically used to address problems such as overdispersion or excessive amounts of zero counts. A typical example in this context

is negative binomial regression, where the prediction function has to be estimated simultaneously with a scale parameter used to model overdispersion. If excessive amounts of zero counts have to be taken into account, it is common to use zero-inflated Poisson or negative binomial models (see Hilbe 2007). With models of this type, the outcome variable is assumed to be a mixture of a zero-generating (Bernoulli) process and a counting process. As a consequence, using zero-inflated Poisson or negative binomial models involves the estimation of a two-dimensional prediction function (where the first component of the prediction function is used to model the zero-generating process and the second component is used to model the counting process). It is important to note that each of the two components may depend on different sets of predictor variables. Furthermore, fitting zero-inflated negative binomial models involves the estimation of an additional scale parameter, where both the two-dimensional prediction function and the scale parameter have to be estimated simultaneously.

Obviously, the examples described above are special cases of a more general estimation problem involving prediction functions with multiple components. We will address this problem by developing a boosting algorithm for multidimensional prediction functions. The proposed algorithm is based on the classical gradient boosting method introduced by Friedman (2001) but is modified such that both parameter estimation and variable selection can be carried out in each component of the multidimensional prediction function. Instead of "descending" the gradient only in one direction, the algorithm computes partial derivatives of the objective function with respect to the various components of the prediction function. In a next step, the algorithm cycles through the partial derivatives, where each component of the prediction function is successively updated in the course of the cycle. This procedure can be modified such that variable selection is carried out in each step of the cycle. If necessary, updates of scale parameters can be obtained at the end of the cycle. This is accomplished by using the current value of the prediction function as an offset value.

As we will demonstrate, the new algorithm constitutes a flexible approach to model fitting in both low-dimensional and high-dimensional data settings. Moreover, the algorithm shares the favorable properties of the classical boosting approach when it comes to efficiency and prediction accuracy. In the special case of a one-dimensional prediction function, the new approach coincides with the original boosting algorithm proposed by Friedman (2001). In addition, it generalizes the work by Schmid and Hothorn (2008b) who developed a boosting algorithm for parametric survival models with a scale parameter. In case of a multinomial logit model, there is a direct correspondence between the new algorithm and the multiclass gradient descent procedure suggested by Hastie et al. (2009).

The rest of the paper is organized as follows: In Section 2, the new algorithm is presented in detail, along with a number of technical details involved in choosing appropriate tuning parameters. The characteristics of the algorithm are demonstrated in Section 3 where an example from epidemiological research is discussed and where the results of a simulation study are presented. A summary of the paper is given in Section 4.

2 Boosting with multidimensional prediction functions

Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be a set of independent realizations of the random variable $(\boldsymbol{X}, \boldsymbol{Y})$, where $\boldsymbol{X}$ is a p-dimensional vector of predictor variables and $\boldsymbol{Y}$ is a (possibly multidimensional) outcome variable. Define $X := (X_1, \ldots, X_n)$ and $Y := (Y_1, \ldots, Y_n)$. The aim is to estimate the K-dimensional prediction function $f^* \in \mathbb{R}^K$ and the L-dimensional set of scale parameters $\sigma^* \in \mathbb{R}^L$, which are defined by

$$\begin{aligned}(f^*, \sigma^*) &= (f_1^*, \ldots, f_K^*, \sigma_1^*, \ldots, \sigma_L^*) \\ &:= \underset{f,\sigma}{\operatorname{argmin}} \, \mathbb{E}_{\boldsymbol{Y},\boldsymbol{X}} \left[\rho(\boldsymbol{Y}, f(\boldsymbol{X}), \sigma) \right] . \end{aligned} \tag{1}$$

The objective function (or "loss function") ρ is assumed to be differentiable with respect to each of the components of $f = (f_1, \ldots, f_K)$.

Usually, in the boosting framework, f^* and σ^* are estimated by minimizing the empirical risk $\sum_{i=1}^n \rho(Y_i, f(X_i), \sigma)$ over f and $\sigma = (\sigma_1, \ldots, \sigma_L)$. Since the components of f^* may have different degrees of complexity, we define a K-dimensional vector of stopping values $m_{\text{stop}} = (m_{\text{stop},1}, \ldots, m_{\text{stop},K})$ that will be used to determine the number of iterations of the boosting algorithm (the choice of m_{stop} will be discussed later). We introduce the following multidimensional extension of the gradient boosting approach developed by Friedman (2001):

1. Initialize the n-dimensional vectors $\hat{f}_1^{[0]}, \ldots, \hat{f}_K^{[0]}$ with offset values, e.g., $\hat{f}_1^{[0]} = \mathbf{0}, \ldots, \hat{f}_K^{[0]} = \mathbf{0}$. Further initialize the one-dimensional scale parameter estimates $\hat{\sigma}_1^{[0]}, \ldots, \hat{\sigma}_L^{[0]}$ with offset values, e.g., $\hat{\sigma}_1^{[0]} = 1, \ldots, \hat{\sigma}_L^{[0]} = 1$. (Alternatively, the maximum likelihood estimates corresponding to the unconditional distribution of $\boldsymbol{Y}$ could be used as offset values.)

2. For each of the K components of f specify a *base-learner*, i.e., a regression estimator with one input variable and one output variable. Set $m = 0$.

3. Increase m by 1.

4. (a) Set $k = 0$.

   (b) Increase k by 1. If $m > m_{\text{stop},k}$ proceed to step 4(f). Else compute the negative partial derivative $-\frac{\partial \rho}{\partial f_k}$ and evaluate at

$$\begin{aligned}\hat{f}^{[m-1]}(X_i) &= \left(\hat{f}_1^{[m-1]}(X_i), \ldots, \hat{f}_K^{[m-1]}(X_i) \right) , \\ \hat{\sigma}^{[m-1]} &= \left(\hat{\sigma}_1^{[m-1]}, \ldots, \hat{\sigma}_L^{[m-1]} \right) ,\end{aligned}$$

$i = 1, \ldots, n$. This yields the negative gradient vector

$$U_k^{[m-1]} = \left(U_{i,k}^{[m-1]}\right)_{i=1,\ldots,n}$$
$$:= \left(-\frac{\partial}{\partial f_k}\rho\left(Y_i, \hat{f}^{[m-1]}(X_i), \hat{\sigma}^{[m-1]}\right)\right)_{i=1,\ldots,n}.$$

(c) Fit the negative gradient vector $U_k^{[m-1]}$ to each of the p components of $\boldsymbol{X}$ (i.e., to each predictor variable) separately by using p times the base-learner (regression estimator) specified in step 2. This yields p vectors of predicted values, where each vector is an estimate of the negative gradient vector $U_k^{[m-1]}$.

(d) Select the component of $\boldsymbol{X}$ which fits $U_k^{[m-1]}$ best according to a pre-specified goodness-of-fit criterion. Set $\hat{U}_k^{[m-1]}$ equal to the fitted values of the corresponding best model fitted in 4(c).

(e) Update $\hat{f}_k^{[m-1]} \leftarrow \hat{f}_k^{[m-1]} + \nu\,\hat{U}_k^{[m-1]}$, where $0 < \nu \leq 1$ is a real-valued step length factor.

(f) For $k = 2, \ldots, K$ repeat steps 4(b) to 4(e). Update $\hat{f}^{[m]} \leftarrow \hat{f}^{[m-1]}$.

5. (a) Set $l = 0$.

(b) Increase l by 1.

(c) Plug $\hat{f}^{[m]}$ and $\hat{\sigma}_1^{[m-1]}, \ldots, \hat{\sigma}_{l-1}^{[m-1]}, \hat{\sigma}_{l+1}^{[m-1]}, \ldots, \hat{\sigma}_L^{[m-1]}$ into the empirical risk function $\sum_{i=1}^n \rho(Y_i, f, \sigma)$ and minimize the empirical risk over σ_l. Set $\hat{\sigma}_l^{[m-1]}$ equal to the newly obtained estimate of σ_l.

(d) For $l = 2, \ldots, L$ repeat steps 5(b) and 5(c). Update $\hat{\sigma}^{[m]} \leftarrow \hat{\sigma}^{[m-1]}$.

6. Iterate Steps 3 to 5 until $m > m_{\text{stop},k}$ for all $k \in \{1, \ldots, K\}$.

From the above algorithm it is easily seen that each component f_k, $k = 1, \ldots, K$, is updated by

1. using the current estimates of the other components $f_1^*, \ldots, f_{k-1}^*$, $f_{k+1}^*, \ldots, f_K^*$ and $\sigma_1^*, \ldots, \sigma_L^*$ as offset values (step 4(b)) and by

2. adding an estimate of the true negative partial derivative $U_k^{[m-1]}$ to the current estimate of f_k^* (step 4(e)).

Note that this strategy is similar to the backfitting strategy developed by Hastie and Tibshirani (1990). With both strategies, components are updated successively by using estimates of the other components as offset values. However, in contrast to gradient boosting (where estimates of f^* are only slightly modified in each iteration), backfitting determines a totally new estimate of f^* in every cycle.

Initialize: $\hat{f}_1^{[0]}, \ldots, \hat{f}_K^{[0]}$ and $\hat{\sigma}_1^{[0]}, \ldots, \hat{\sigma}_L^{[0]}$ with offset values.
  for $k = 1$ to K **do**
    Specify a base-learner for component f_k.
  end for
Evaluate:
  for $m = 1$ to $\max(m_{\text{stop}})$ **do**
    for $k = 1$ to K **do**
      if $m \leq m_{\text{stop},k}$ **then**
        (i) Compute $-\frac{\partial \rho}{\partial f_k}$ and evaluate at $\hat{f}^{[m-1]}(X_i)$, $\hat{\sigma}^{[m-1]}$, $i = 1, \ldots, n$. This yields $U_k^{[m-1]}$.

        (ii) Fit $U_k^{[m-1]}$ to each of the p components of $\boldsymbol{X}$ separately by using p times the base-learner.

        (iii) Select the component of $\boldsymbol{X}$ which fits $U_k^{[m-1]}$ best. Set $\hat{U}_k^{[m-1]}$ equal to the fitted values from the best-fitting model.

        (iv) Update $\hat{f}_k^{[m-1]} \leftarrow \hat{f}_k^{[m-1]} + \nu \hat{U}_k^{[m-1]}$.
      end if
    end for
    Update $\hat{f}^{[m]} \leftarrow \hat{f}^{[m-1]}$.
    for $l = 1$ to L **do**
      Plug $\hat{f}^{[m]}$ and $\hat{\sigma}_1^{[m-1]}, \ldots, \hat{\sigma}_{l-1}^{[m-1]}, \hat{\sigma}_{l+1}^{[m-1]}, \ldots, \hat{\sigma}_L^{[m-1]}$ into the empirical risk function and minimize over σ_l. Set $\hat{\sigma}_l^{[m-1]}$ equal to the newly obtained estimate of σ_l^*.
    end for
    Update $\hat{\sigma}^{[m]} \leftarrow \hat{\sigma}^{[m-1]}$.
  end for

Figure 1: Gradient boosting with multidimensional prediction functions.

After having obtained an update of $\hat{f}$ in step 4, the algorithm cycles through the components of σ, where in each step of the cycle, an update of σ_l, $l = 1, \ldots, L$, is obtained (step 5). This is accomplished by minimizing the empirical risk (evaluated at the current estimates of the other parameters f^* and $\sigma_1^*, \ldots, \sigma_{l-1}^*, \sigma_{l+1}^*, \ldots, \sigma_L^*$) numerically. A summary of the algorithm is given in Figure 1.

The values of the stopping iterations $m_{\text{stop},1}, \ldots, m_{\text{stop},K}$ are the main tuning parameters of the algorithm. In case of classical gradient boosting with only one dimension $K = 1$, it has been argued that boosting algorithms should not be run until convergence. Otherwise, overfits resulting in suboptimal prediction rules would be likely (see Bühlmann and Hothorn 2007). Usually, cross-validation (CV) techniques are used to determine the value of $m_{\text{stop},1}$ (i.e., $m_{\text{stop},1}$ is the iteration with lowest predictive risk). In case of the new boosting algorithm with K dimensions, we will use K-dimensional cross-validation,

i.e., the predictive risk is evaluated on a K-dimensional grid corresponding to combinations of $m_{\text{stop},1}, \ldots, m_{\text{stop},K}$. This strategy will be further discussed in Section 3. In principle, many types of cross-validation techniques can be applied to estimate $m_{\text{stop},1}, \ldots, m_{\text{stop},K}$ (leave-one-out CV, k-fold CV, repeated k-fold CV, bootstrap CV with 0.632 or 0.632+ adjustments, etc.). While it is possible that different cross-validation techniques may have different effects on the estimates of the stopping iterations, we will restrict ourselves to using five-fold cross-validation in this paper.

The choice of the step length factor ν has been shown to be of minor importance with respect to the predictive performance of the classical boosting algorithm ($K = 1$). The only requirement is that the value of ν is small ($0 < \nu \leq 0.1$), such that a stagewise adaption of the true prediction function is possible (see Bühlmann and Hothorn 2007 or Schmid and Hothorn 2008a). In the remainder of the paper, a constant value of ν ($= 0.1$) will be used for all K dimensions. In step 4(d) of the algorithm we will use the R^2 measure of explained variation as the goodness-of-fit criterion (since the vectors $U_k^{[m-1]}$ are measured on a continuous scale).

As outlined in Section 1, the algorithm combines model estimation with the selection of the most relevant predictor variables. It is important to note that the "component-wise" variable selection mechanism introduced in step 4 of the algorithm is not a necessary feature of boosting algorithms (since, in principle, any type of regression model could be used to compute estimates of the negative gradient). However, incorporating variable selection into boosting algorithms typically increases the efficiency of model estimates and predictions, especially if $p > n$ (see, e.g., Bühlmann and Yu 2003 or Bühlmann and Hothorn 2007).

In steps 4(c) to 4(e), by using a regression estimator as the base-learner, a structural relationship between $\boldsymbol{Y}$ and the set of predictors $\boldsymbol{X}$ is established. Due to the additive structure of the update (step 4(e)), the final estimates of $f_1^*, \ldots, f_K^*$ at iterations m_{stop} are fits of an additive model but will depend only on a subset of the p components of $\boldsymbol{X}$. In each iteration, the algorithm selects the basis direction "closest" to the descent direction of the prediction function (step 4(d)). Since only one element of $\boldsymbol{X}$ is used for updating the prediction function in step 4(e), the algorithm is applicable even if $p > n$. The step length factor ν can be viewed as a regularization factor used for shrinking the predictions $\hat{f}^{[m]}$. In this context, the proposed algorithm can be interpreted as a "stagewise regression" technique (cf. Efron et al. 2004).

It is easily seen that in case of a one-dimensional prediction function $f^* \equiv f_1^*$ and an empty set of scale parameters, the boosting algorithm presented above reduces to the classical gradient descent algorithm developed by Friedman (2001). Similarly, if $f^* \equiv f_1^*$ and σ^* is one-dimensional, the algorithm is a generalization of the model fitting approach developed by Schmid and Hothorn (2008b) (where boosting was used for deriving parametric survival prediction rules). In case of a multinomial logit model with K outcome categories, the

conditional probability of falling into category k is typically modeled via

$$\mathrm{P}\left(\boldsymbol{Y} = k|\boldsymbol{X}\right) = \frac{e^{f_k^*(\boldsymbol{X})}}{\sum_{j=1}^{K} e^{f_j^*(\boldsymbol{X})}} \ . \tag{2}$$

Thus, by setting σ^* equal to the empty set and $m_{\text{stop},1} = \ldots = m_{\text{stop},K}$, the boosting algorithm introduced above can be used for fitting the multiclass model defined by (2). If the negative multinomial log likelihood

$$\rho_{\text{multinom}}(Y, f) = -\sum_{i=1}^{n}\sum_{k=1}^{K} \mathrm{I}(Y_i = k) f_k(X_i) + \sum_{i=1}^{n} \log\left(\sum_{j=1}^{K} e^{f_j(X_i)}\right) \tag{3}$$

is used as the loss function, the proposed algorithm will give essentially the same result as the multiclass method suggested by Hastie et al. (2009), Algorithm 10.4.

Finally, the boosting algorithm presented above can easily be modified such that the components of f are restricted to depend on subsets $\chi_1, \ldots, \chi_K \subset \{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p\}$ only. Reducing the predictor spaces in step 4 of the algorithm adds considerable flexibility to the boosting procedure, since it allows for taking into account prior knowledge about the dependency between f_k^* and $\boldsymbol{Y}$.

3 Examples

3.1 Modeling nevus counts of preschool children

Nevus counts of children have been established as an important risk factor for malignant melanoma occurring in later life (Gallagher et al. 1990). To address this issue in an epidemiologic study, the CMONDE Study Group (Uter et al. 2004, Pfahlberg et al. 2004) conducted a standardized skin assessment of consecutive cohorts of preschool children in the German town of Göttingen. Nevus counts were collected in the course of a mandatory medical examination prior to school enrollment in 1999 and 2000. For reasons of homogeneity we focus here on the subset of $n = 1235$ children examined in 1999. Predictor variables in the data set included three continuous predictors (age, skin pigmentation, body mass index) and five categorical predictors (sex, hair color, skin type, color of iris, degree of freckling). The number of possible combinations of the categories was equal to 576.

In the following we will use the eight predictor variables to model expected nevus counts of children. In order to construct accurate predictions of the nevus counts, identification of relevant covariates is necessary. Also, given the fact that a relatively large number of categories is involved in the modeling process, some sort of regularization of the prediction function is desirable. Since the algorithm introduced in Section 2 can be used to obtain both regularized estimates and an interpretable prediction function, it is an appropriate technique for addressing these issues.

We will compare the prediction accuracy of boosting estimates obtained from four different loss functions, where each loss function corresponds to a particular type of count data model:

1. *Negative Poisson log likelihood loss.* The most popular distribution used for modeling count data is the Poisson distribution. In the generalized linear model (GLM) setting (McCullagh and Nelder 1989), Poisson model estimates are obtained by maximizing the conditional log likelihood

$$\begin{aligned} l_{\mathrm{Po}}(Y, f_1) &= \sum_{i=1}^{n} Y_i \cdot f_1(X_i) \\ &\quad - \sum_{i=1}^{n} \Big(\log(Y_i!) + \exp(f_1(X_i)) \Big) \end{aligned} \tag{4}$$

   over a one-dimensional prediction function $f = f_1$ (where $\exp(f_1(\boldsymbol{X}))$ corresponds to the conditional expectation of the outcome variable $\boldsymbol{Y}$ given the predictors $\boldsymbol{X}$). Since maximum likelihood estimation tends to become unstable in the presence of a larger number of categorical predictors, we use the boosting algorithm introduced in Section 2 for obtaining estimates of the optimal prediction function f^*. This is accomplished by setting the loss function ρ equal to the negative Poisson log likelihood and the set of scale parameters σ equal to the empty set.

2. *Negative NB log likelihood loss.* The underlying assumptions of the Poisson model are often too restrictive for capturing the full variability contained in a data set. A common way to model such overdispersed data is to consider negative binomial (NB) regression models. The log likelihood of the negative binomial model is given by

$$\begin{aligned} l_{\mathrm{NB}}(Y, f_1, \sigma_1) &= \sum_{i=1}^{n} \Big(\log[\Gamma(Y_i + \sigma_1)] - \log(Y_i!) \Big) \\ &\quad - n \log[\Gamma(\sigma_1)] + \sum_{i=1}^{n} Y_i \cdot f_1(X_i) \\ &\quad + \sum_{i=1}^{n} \sigma_1 \log\left(\frac{\sigma_1}{\exp(f_1(X_i)) + \sigma_1} \right) \\ &\quad - \sum_{i=1}^{n} Y_i \log(\exp(f_1(X_i)) + \sigma_1) \ , \end{aligned} \tag{5}$$

   where $f = f_1$ is a one-dimensional prediction function and $\sigma = \sigma_1$ is a one-dimensional scale parameter used for modeling the variance of $\boldsymbol{Y}$. It is well known that $\lambda := \exp(f_1(\boldsymbol{X}))$ corresponds to the conditional expectation of the outcome variable $\boldsymbol{Y}$ given the predictors $\boldsymbol{X}$, and that the conditional

variance of $\boldsymbol{Y}|\boldsymbol{X}$ is given by $\lambda + \lambda^2/\sigma_1$. The log likelihood given in (5) converges to the Poisson log likelihood as $\sigma_1 \to \infty$. In the following we will use the boosting algorithm introduced in Section 2 for obtaining estimates of the optimal parameters f_1^* and σ_1^*. This is accomplished by setting ρ equal to the negative NB log likelihood and σ equal to the scale parameter σ_1 in (5).

3. *Negative zero-inflated Poisson log likelihood loss.* Excessive amounts of zero counts, i.e., more zeros than expected in a Poisson or negative binomial model, are a common problem associated with count data. In case of the CMONDE data, the fraction of zero nevus counts is approximately 9.2%, which is about 20 times as much as the corresponding fraction to be expected from the unconditional Poisson distribution of the nevus counts (0.479%). In order to take this problem into account, we additionally fit a zero-inflated Poisson model to the CMONDE data. The log likelihood of the zero-inflated Poisson model is given by

$$\begin{aligned} l_{\text{ZIPo}}(Y, f_1, f_2) \;=\; & - \sum_{i:Y_i=0} \log\left(1 + e^{f_1(X_i)}\right) \\ & + \sum_{i:Y_i=0} \log\left(e^{f_1(X_i)} + e^{-e^{f_2(X_i)}}\right) \\ & - \sum_{i:Y_i>0} \log\left(1 + e^{f_1(X_i)}\right) \\ & + \sum_{i:Y_i>0} \left(Y_i \cdot f_2(X_i) - \log\left(Y_i!\right)\right) \\ & - \sum_{i:Y_i>0} e^{f_2(X_i)} \;, \end{aligned} \tag{6}$$

where f_1 is the predictor of the binomial logit model

$$\mathrm{P}\left(\boldsymbol{Z} = 0|\boldsymbol{X}\right) = \frac{e^{f_1(\boldsymbol{X})}}{1 + e^{f_1(\boldsymbol{X})}} \tag{7}$$

with binary outcome variable $\boldsymbol{Z} \in \{0, 1\}$, and f_2 is the predictor of the Poisson model

$$\mathrm{P}\big(\tilde{\boldsymbol{Y}} = k|\boldsymbol{X}\big) = \frac{e^{k \cdot f_2(\boldsymbol{X})}}{k!}\, e^{-e^{f_2(\boldsymbol{X})}} \tag{8}$$

with Poisson-distributed outcome variable $\tilde{\boldsymbol{Y}}$. It is easily seen from (6) to (8) that the zero-inflated Poisson model is a mixture of a point mass at zero (accounting for an extra amount of zeros) and a Poisson distribution. For details we refer to Hilbe (2007). Since we want to regularize the estimates of both components of the prediction function, we use the boosting algorithm introduced in Section 2. This is achieved by setting ρ equal to the negative log likelihood given in (6) and $f = (f_1, f_2)$. The set of scale parameters σ is set equal to the empty set.

4. *Negative zero-inflated NB log likelihood loss.* In case of overdispersed data, modeling additional amounts of zero counts can be accomplished by using the zero-inflated negative binomial model. The log likelihood of this model is given by

$$
\begin{aligned}
l_{\text{ZINB}}(Y, f_1, f_2, \sigma_1) = &- \sum_{i=1}^{n} \log\left(1 + e^{f_1(X_i)}\right) \\
&+ \sum_{i:Y_i=0} \log\left(e^{f_1(X_i)} + \left(\frac{e^{f_2(X_i)} + \sigma_1}{\sigma_1}\right)^{-\sigma_1}\right) \\
&- \sum_{i:Y_i>0} \sigma_1 \log\left(\frac{e^{f_2(X_i)} + \sigma_1}{\sigma_1}\right) \\
&- \sum_{i:Y_i>0} Y_i \log\left(1 + e^{-f_2(X_i)} \cdot \sigma_1\right) \\
&- \sum_{i:Y_i>0} \Big(\log\left(\Gamma(\sigma_1)\right) + \log\left(\Gamma(1 + Y_i)\right)\Big) \\
&+ \sum_{i:Y_i>0} \log\left(\Gamma(\sigma_1 + Y_i)\right) \ . \qquad (9)
\end{aligned}
$$

   Similar to the zero-inflated Poisson model, the zero-inflated negative binomial model is a mixture of a point mass at zero (modeled by a binomial GLM) and a zero-inflated negative binomial regression model. We apply the new boosting algorithm to the CMONDE data by setting ρ equal to the negative log likelihood given in (9) and $f = (f_1, f_2)$. The set of scale parameters σ is set equal to the scale parameter σ_1 in (9).

In order to compare the four models described above, we carried out a benchmark study using the CMONDE data. In a first step, the full data set was randomly split 50 times into pairs of training samples and test samples. Each training sample contained 1111 observations, i.e., about 90% of the data. In a next step, the boosting algorithm introduced in Section 2 was used to estimate the parameters of the four count data models. As base-learners, simple *linear* regression models were used, so that the components of $\hat{f}$ became linear functions of the predictors. As a consequence of this strategy, coefficient estimates were obtained for each predictor variable. For all components of f, variables were selected from the full set of predictors (i.e., no restrictions were made to the set of predictors at the beginning of the algorithm). In a last step, the prediction rules obtained from the four models were evaluated using the 50 test samples. All computations were carried out with the R System for Statistical Computing (version 2.7.2, R Development Core Team 2008) using a modification of the `glmboost()` function in package **mboost** (version 1.0-4, Hothorn et al. 2008). In case of the Poisson and negative binomial models, we ran five-fold cross-validation on the training samples to determine the 50 values of the stopping iteration $m_{\text{stop},1}$. In case of the zero-inflated Poisson and negative binomial models, this approach had to be extended: Since the prediction

functions of these two models are two-dimensional, it is possible for the components of the prediction functions to have different degrees of complexity. To take this issue into account, we ran two-dimensional five-fold cross-validation on the 50 training samples, determining pairs of optimal stopping iterations $m_{\text{stop},1}$ and $m_{\text{stop},2}$.

Since the negative versions of the log likelihood functions (4), (5), (6) and (9) are used as loss functions for the respective boosting algorithms, it would be a natural approach to measure the prediction accuracy of the boosting methods by computing the predictive log likelihood values from the test samples. Since the functions (4), (5), (6) and (9) are measured on different scales, however, using this approach would be unsuitable for comparing the four models. We therefore used the Brier score (Brier 1950), which is a model-independent measure of prediction error. The Brier score is defined as the average squared distance between the observed proportions and the predicted proportions of the outcome categories in the test samples. Thus, since the count data models under consideration are characterized by equidistant outcome categories, using the Brier score corresponds to using the integrated squared difference of the predicted and observed c.d.f.'s of the test observations as a measure of prediction error.

More formally, the Brier score for test sample t, $t \in \{1, \ldots, 50\}$, is defined as

$$BS_t := \frac{1}{n_t} \sum_{k=1}^{M} \sum_{i=1}^{n_t} \left(p_{ikt} - \hat{p}_{ikt}\right)^2 \ , \tag{10}$$

where n_t is the number of observations in test sample t and M is the number of categories of the outcome variable. Let Y_{it} be the i-th realization of the outcome variable $\boldsymbol{Y}$ in test sample t. Then the parameters p_{ikt} in (10) are defined as

$$p_{ikt} = \begin{cases} 1 & \text{if } Y_{it} = k \\ 0 & \text{otherwise} \end{cases} \ . \tag{11}$$

The parameters p_{ikt} can be interpreted as the observed proportions of category k given X_{it} (where X_{it} denotes the i-th realization of the predictor variables $\boldsymbol{X}$ in test sample t). Similarly, the predicted proportions of category k given X_{it} (denoted by $\hat{p}_{ikt}$) are obtained by plugging the estimates of f^* and σ^* (computed from *training* sample t) into the log likelihood functions corresponding to test sample t. For computational reasons we defined $p_{iMt} := 1 - \sum_{k \leq M^*} p_{ikt}$, where M^* is the largest outcome value observed in the CMONDE data.

The Brier score can generally be used for assessing the quality of probabilistic forecasts. It is an example of a so-called "proper" scoring rule, where "proper" means that the expectation of (10) is minimized if the predictions $\hat{p}_{ikt}$ are computed from the true model with parameters f^* and σ^*. For details on proper scoring rules we refer to Gneiting and Raftery (2007). Generally, a small value of the Brier score corresponds to a highly accurate prediction rule (and vice versa).

In Figure 2, boxplots of the Brier score values computed from the 50 test samples of the CMONDE data are shown. In addition, Figure 2 includes the Brier score values corresponding to a Poisson "null" model with no covariate

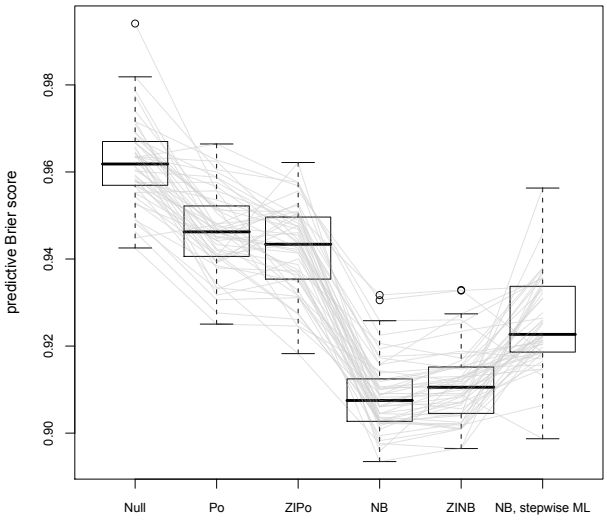

Figure 2: Predictive Brier score values computed from the 50 test samples of the CMONDE data (Null = null model, Po = Poisson model, ZIPo = zero-inflated Poisson model, NB = negative binomial model, ZINB = zero-inflated negative binomial model, NB, stepwise ML = predictions obtained from maximum likelihood estimation of the negative binomial model). Obviously, the negative binomial models perform better than the Poisson models, indicating that overdispersion is present in the CMONDE data. In case of the negative binomial model, using boosting estimates instead of maximum likelihood estimates results in an increase of prediction accuracy.

information. Obviously, in most cases, the Brier score values corresponding to the two Poisson regression models are smaller than those corresponding to the null model. It can also be seen from Figure 2 that introducing a scale parameter for modeling overdispersion, i.e., using a negative binomial model, leads to improved predictions. This result indicates that not all of the structure contained in the data is captured by the Poisson distribution. If compared to the negative binomial model, the zero-inflated negative binomial model does not seem to lead to an additional increase in predictive power. The negative binomial model even results in smaller prediction errors on average than its zero-inflated version. This conjecture is confirmed by a Wilcoxon signed rank test on the differences between the Brier score values of the two models ($p = 0.001$). Therefore, and also because of its simpler structure, we suggest to use the negative binomial model for predicting nevus counts.

The coefficient estimates of the negative binomial model (computed from the full data set) are shown in Table 1. When comparing the directions of the estimates to the results published by the CMONDE study group, it becomes obvious that the original results (Pfahlberg et al. 2004) are supported by the boosting estimates. For example, children with blonde hair tend to have substantially more nevi than children with black hair. Also, the number of facial freckles is positively correlated with the number of nevi, where male children have more nevi (on average) than female children. A detailed description of the variables contained in Table 1 and their effect on nevus counts can be found in Pfahlberg et al. (2004). The selection rates of the eight covariates are shown in Table 2. Obviously, selection rates corresponding to the count components are higher on average than selection rates corresponding to the zero components. While age and body mass index do not seem to have a large effect on the zero component of the zero-inflated models, skin pigmentation and facial freckles seem to explain many of the observed zero counts. In case of the negative binomial model, selection rates of covariates are generally very high. This result suggests that all eight covariates are important if a negative binomial model is used for prediction of nevus counts. Obviously, in case of the negative binomial, it is the shrinkage property of the boosting method that leads to an optimized prediction accuracy.

In a last step, we used the 50 training samples to compute Brier score values from the "classical" negative binomial maximum likelihood (ML) estimates. Here, no shrinkage procedures were applied, and variable selection was carried out in a stepwise fashion (using the AIC as a goodness-of-fit criterion). Figure 2 illustrates that in case of the CMONDE data, the regularization properties of boosting are indeed superior to those of ML estimation with stepwise variable selection. This result demonstrates that if predictive accuracy and estimation accuracy are considered to be equally important, boosting is a very good alternative to ML estimation even in low-dimensional settings. However, average prediction accuracy is still higher in case of the negative binomial ML estimates than in case of the boosting estimates obtained from the Poisson models. This result suggests that regularization cannot fully overcome problems caused by choosing a wrong type of count data model. Very similar results were obtained when the BIC was used as a goodness-of-fit criterion for stepwise variable selection in the ML setting.

3.2 Simulation study on zero-inflated negative binomial regression

In this section we discuss the estimation and regularization properties of the proposed boosting algorithm in high-dimensional settings ($p > n$). We will analyze the boosting estimates obtained from a simulation study on zero-inflated

Table 1: Boosting coefficient estimates obtained from the CMONDE data (negative binomial model). The 95% intervals correspond to the 2.5 and 97.5 percentiles computed from 500 bootstrap samples. Skin type was determined using the categories proposed by Fitzpatrick (1988). Skin pigmentation was quantified with remission photometry, i.e., small reflectance measurements correspond to a highly pigmented skin.

Predictor	Est. Coef.	95% interval
Intercept	-0.4400	[-0.4598, -0.0977]
sex (male)	0.0000	
sex (female)	-0.0350	[-0.1142, 0.0387]
hair color (blonde)	0.0000	
hair color (brown)	-0.0497	[-0.1644, 0.0132]
hair color (red)	-0.6253	[-1.2318, -0.0339]
hair color (black)	-0.3432	[-0.8132, -0.0661]
Fitzpatrick skin type (I)	0.0000	
Fitzpatrick skin type (II)	0.2648	[0.0159, 0.3964]
Fitzpatrick skin type (III)	0.1557	[-0.0709, 0.2863]
Fitzpatrick skin type (IV)	-0.0288	[-0.2600, 0.0327]
color of iris (blue)	0.0000	
color of iris (dark brown)	-0.5714	[-0.8634, -0.2400]
color of iris (green-blue)	0.0191	[-0.1273, 0.1561]
color of iris (green-brown)	0.0019	[-0.1719, 0.1634]
color of iris (light blue)	-0.0317	[-0.2140, 0.0722]
color of iris (light brown)	-0.1620	[-0.3202, 0.0000]
facial freckles (none)	0.0000	
facial freckles (few)	0.1547	[0.0269, 0.2848]
facial freckles (many)	0.1766	[-0.0344, 0.3945]
skin pigmentation (reflectance in % at 650 nm)	0.0240	[0.0010, 0.0454]
age in years	0.0932	[0.0000, 0.2268]
body mass index in kg/m^2	0.0331	[0.0154, 0.0534]
$\hat{\sigma}$	1.8303	[1.7477, 2.0664]

negative binomial regression with $\sigma_1 = 3$ and with linear prediction functions

$$\begin{aligned} f_1 &= \mathbf{X}^\top \beta \\ &= -0.4 \cdot \mathbf{X}_1 - 0.2 \cdot \mathbf{X}_2 + 0 \cdot \mathbf{X}_3 + 0.2 \cdot \mathbf{X}_4 + 0.4 \cdot \mathbf{X}_5 \\ &\quad \text{(zero component)}, \end{aligned}$$

$$\begin{aligned} f_2 &= \mathbf{X}^\top \gamma \\ &= 0.4 \cdot \mathbf{X}_1 + 0.2 \cdot \mathbf{X}_2 + 0 \cdot \mathbf{X}_3 - 0.2 \cdot \mathbf{X}_4 - 0.4 \cdot \mathbf{X}_5 \\ &\quad \text{(count component)}, \end{aligned}$$

Table 2: Selection rates of covariates (in %) obtained from the 50 training samples of the CMONDE data (Po = Poisson model, ZIPo = zero-inflated Poisson model, NB = negative binomial model, ZINB = zero-inflated negative binomial model, CC = count component, ZC = zero component). In case of the two zero-inflated models, selection rates corresponding to the count components are higher on average than selection rates corresponding to the zero components. Apart from indicating that the two components have different degrees of complexity, this result also suggests that many of the zero counts observed in the CMONDE data can solely be explained by the count components of the zero-inflated models.

	Po, CC	ZIPo, CC	ZIPo, ZC
sex	100	100	12
hair color	100	100	44
Fitzpatrick skin type	100	100	54
color of iris	100	100	48
facial freckles	100	100	60
skin pigmentation	10	80	100
age	44	96	0
bmi	100	100	12
	NB, CC	ZINB, CC	ZINB, ZC
sex	92	100	2
hair color	100	100	46
Fitzpatrick skin type	100	100	78
color of iris	100	100	74
facial freckles	100	100	88
skin pigmentation	100	22	100
age	94	2	0
bmi	100	26	0

where $\beta := (\beta_1, \ldots, \beta_5)^\top = (-0.4, -0.2, 0, 0.2, 0.4)^\top$, $\gamma := (\gamma_1, \ldots, \gamma_5)^\top = (0.4, 0.2, 0, -0.2, -0.4)^\top$, and where the covariates $\mathbf{X}_j$, $j = 1, \ldots, 5$, followed a normal distribution with zero mean and standard deviation sd $= \sqrt{5}$ each. With f_1, f_2 and σ_1 taking the above values, the dependent variable of the zero-inflated negative binomial model will have approximately 67% zero counts on average. To simulate high-dimensional data settings, we added 1000 non-informative covariates $\mathbf{X}_6, \ldots, \mathbf{X}_{1005}$ to the covariate space. Each of the additional covariates followed a normal distribution with zero mean and standard deviation sd $= \sqrt{5}$. All covariates (including $\mathbf{X}_1, \ldots, \mathbf{X}_5$) were equicorrelated with correlation coefficient $\rho = 0.5$. Note that covariate $\mathbf{X}_3$ is also non-informative. It will serve as an example of a non-informative predictor and will be studied in more detail than the other non-informative predictors (see Figure 3).

Since the prediction functions f_1 and f_2 are linear in the predictors, simple linear regression models were used as base-learners in the proposed boosting algorithm. Three values of n were considered ($n = 400, 600, 800$). For each value of n, boosting was applied to 50 independent data sets generated from the zero-inflated negative binomial regression model defined above. The values of the stopping iterations $m_{\text{stop},1}$ and $m_{\text{stop},2}$ were determined by evaluating independent test data sets of size $n_{\text{test}} = 1000$ that were generated from the same model as the original 50 data sets.

In Figure 3, boxplots of the 50 parameter estimates of β and γ are shown. As expected, all coefficient estimates are shrunken towards zero. The signs of the coefficient estimates (and also the magnitudes of the coefficient estimates relative to each other) clearly reflect the true structures of f_1 and f_2. Selection rates of non-informative covariates are close to zero for all n (see Table 3). Table 3 also reveals that in case of a small sample size ($n = 400$), there is still a considerable amount of samples where informative covariates have not been selected. This result, which is especially true for the zero component f_1, indicates that different components of a prediction function may require different degrees of shrinkage in order to optimize the overall predictive power of the model. Despite the predictive power being maximized, the coefficient estimates obtained for $n = 400$ clearly lack interpretability (since in many cases informative covariates have not been selected). As n increases, coefficient estimates become larger and selection rates of informative covariates increase. For $n = 800$, boosting seems to work reasonably well with respect to both variable selection and interpretability of coefficient estimates. In this context, it is important to note that a sample size of $n = 800$ is still smaller than typical sample sizes needed for obtaining stable maximum likelihood estimates in low-dimensional settings. As an example, consider the data sets that have been used by Hilbe (2007) to demonstrate maximum likelihood estimation of zero-inflated regression models. In case of these data sets, n is always larger than 1000 while $p \leq 5$.

We finally analyzed the behavior of the stopping iterations $m_{\text{stop},1}$ and $m_{\text{stop},2}$ when the components of the prediction function have varying degrees of complexity. To do this, we kept the value of $\gamma = (0.4, 0.2, 0, -0.2, -0.4)^\top$ constant and considered three different settings for β:

$$\begin{aligned} \text{setting 1: } \beta &= \mathbf{0}\,, \\ \text{setting 2: } \beta &= -\gamma\,, \\ \text{setting 3: } \beta &= -2 \cdot \gamma\,. \end{aligned}$$

In setting 2, both components f_1 and f_2 have the same complexity with respect to β and γ. In setting 1, the complexity of f_1 is smaller than the complexity of f_2 while in setting 3 it is "twice" as large. We expect the estimates of the optimal values $m_{\text{stop},1}$ to increase with increasing complexity of f_1.

Table 4 shows the results of a simulation study with 20 simulation runs and $n = 600$. Stopping was performed in the same way as in the previous simulation study. While $m_{\text{stop},2}$ is relatively constant on average in all three settings, $m_{\text{stop},1}$ indeed increases with increasing complexity of f_1. Moreover,

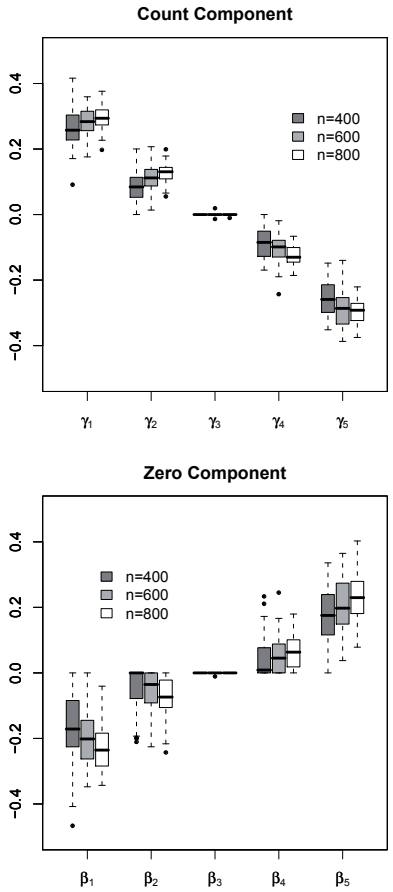

Figure 3: Coefficient estimates corresponding to the 50 samples generated from the zero-inflated negative binomial model defined in Section 3.2. Obviously, the magnitude of the coefficient estimates increases with n increasing. In case of the non-informative covariate $\mathbf{X}_3$, selection frequencies are zero for almost all samples. Consequently, the estimates of γ_3 and β_3 are also equal to zero in these cases.

we see from Figure 4 that a two-dimensional evaluation of the stopping iterations is necessary for minimizing prediction errors when the complexities of f_1 and f_2 are different: If the same stopping iteration was used for both components (i.e., when $m_{\text{stop},1}$ was restricted to be equal to $m_{\text{stop},2}$), the predictive risk increased in comparison to the two-dimensional stopping approach.

Table 3: Selection rates of covariates (in %) obtained from the 50 samples generated from the zero-inflated negative binomial regression model defined in Section 3.2. Obviously, selection rates increase with n increasing. In case of the zero component of the model, selection rates of informative covariates are small for $n = 400$. The last column contains the selection rates corresponding to the non-informative covariates $\mathbf{X}_6 - \mathbf{X}_{1005}$. In case of the latter covariates, variable selection works remarkably well.

count component						
	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	$\mathbf{X}_5$	$\mathbf{X}_6 - \mathbf{X}_{1005}$
$n = 400$	100	94	0	86	100	2.73
$n = 600$	100	100	4	100	100	2.95
$n = 800$	100	100	2	100	100	3.37
zero component						
	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	$\mathbf{X}_5$	$\mathbf{X}_6 - \mathbf{X}_{1005}$
$n = 400$	98	46	0	56	96	1.18
$n = 600$	98	58	2	68	100	1.17
$n = 800$	100	84	0	82	100	1.48

4 Summary and conclusion

Originally developed as a machine learning technique for predicting binary outcomes (Freund and Schapire 1997), boosting has gained considerable attention in the statistical community over that last years. Most notably, by showing that the original boosting algorithm for binary classification can be interpreted as a gradient descent technique for minimizing arbitrary loss functions, Breiman (1998, 1999) has laid the foundations for applying boosting algorithms to a wide class of statistical estimation problems. Later, by introducing the "statistical view" of boosting, Friedman et al. (2000) have established boosting as a tool for fitting very general types of regression models. Due to their regularization properties, boosting algorithms can generally be used to achieve a balance between estimation and prediction accuracy (see Friedman et al. 2000; Bühlmann and Hothorn 2007).

In this paper we have extended the classical gradient boosting approach by constructing a boosting algorithm for regression models with multidimensional prediction functions. Instead of descending the gradient in one direction only, the algorithm introduced in this paper successively computes the partial derivatives of the components of the prediction function. Updates of a component of the prediction function are then computed by using the current values of the other components as offset values. Most important, the regularization concept of the original boosting approach carries over to the new multidimensional algorithm. As a result, the algorithm introduced in this paper is useful for analyzing

Table 4: Average stopping iterations $m_{\text{stop},1}$ (line 1) and $m_{\text{stop},2}$ (line 2) obtained from 20 samples of a zero-inflated negative binomial regression model. The three parameter settings defined in Section 3.2 were considered ($n = 600$, ZC = zero component, CC = count component). Standard deviations are given in brackets. Obviously, the average value of $m_{\text{stop},1}$ increases with increasing complexity of f_1. Since the complexity of f_2 is equal in all three settings, the average values of $m_{\text{stop},2}$ are relatively constant.

	setting 1	setting 2	setting 3
ZC, f_1	160 (88.26)	665 (397.72)	3085 (1071.29)
CC, f_2	550 (274.34)	450 (216.43)	395 (119.10)

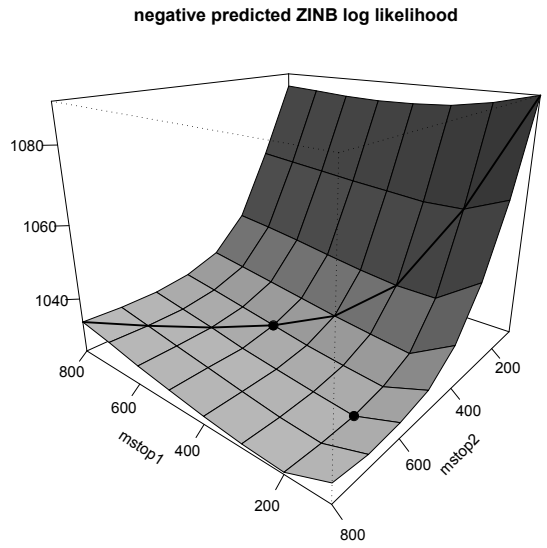

Figure 4: Predictive risk obtained from the first simulation run in setting 1 (Section 3.2). Obviously, two different stopping iterations are needed for minimizing the predictive risk. The two-dimensional stopping strategy improves prediction accuracy if compared to a one-dimensional stopping strategy with the same number of iterations for each component of the prediction function. This result could be observed in almost all simulation runs.

both low-dimensional and high-dimensional data (where selecting a moderate number of relevant predictors is often a key problem).

As demonstrated in Section 3, boosting with multidimensional prediction functions is a suitable technique for fitting count data models with different types of prediction functions and scale parameters. Apart from the models considered in this paper, the algorithm could easily be used to fit other popular types of count data models, such as the Hurdle model (Mullahy 1986) or the generalized Poisson distribution (Consul and Jain 1973).

Concerning the regularization properties of the proposed boosting algorithm, our empirical results suggest that in order to optimize prediction accuracy, different components of a multidimensional prediction function may require different degrees of regularization. As a consequence, we recommend to use a multidimensional cross-validation strategy for determining the values of the stopping iterations. Clearly, this approach might become infeasible if the number of components of the prediction function is too large. In such situations, defining groups of related components and applying the same number of boosting iterations to the components of each group could be a viable strategy. For example, in case of the gradient boosting algorithm for multiclass prediction, Hastie et al. (2009) used the same stopping iteration for all outcome categories.

In contrast to the values of the stopping iterations, the value of the step length factor ν does not seem to have a large effect on prediction accuracy. It is, however, important to keep ν small ($\nu \leq 0.1$). Additional experiments that were conducted in the course of our simulation study revealed that the proposed algorithm may become instable if $\nu > 0.5$. This is, however, also true for the classical boosting algorithm with a one-dimensional prediction function.

In addition to the count data examples presented in Section 3, the proposed algorithm is generally suitable for solving a wide class of estimation problems with multidimensional prediction functions. In particular, boosting constitutes a natural approach to estimating the parameters of identifiable finite mixture models, where, in addition to the regression parameters, the class probabilities of a fixed number of latent categories have to be estimated. (Note that the zero-inflated count data models considered in Section 3 are special cases of finite mixture models.) Furthermore, the algorithm can easily be modified such that different types of base-learners can be applied to different components of the prediction function. For example, one could use smooth base-learners to model the first component of the prediction function, tree base-learners to model the second component, linear base-learners to model the third component, etc. Similarly, by using two-dimensional base-learners, interaction terms between the covariates can be included into the prediction function. In addition, instead of updating scale parameters by numerical optimization, it is possible to regress them on a (possibly restricted) set of covariates. This can easily be accomplished by treating the (sub)set of scale parameters as an ordinary component of the prediction function, i.e., by modeling the scale parameter(s) via the same base-learning procedures as used for the prediction function. In this regard, boosting with multidimensional prediction functions constitutes a highly flexible approach to statistical model estimation.

Acknowledgements

The authors thank the Regional Health Authority of the city of Göttingen (Head: Dr. W. R. Wienecke) and Prof. Dr. K. F. Kölmel (Department of Dermatology at the University of Göttingen) for the permission to use the data of the CMONDE study for illustrative purposes. The CMONDE study was supported by the Cancer Fund of Lower Saxony. MS and TH were supported by Deutsche Forschungsgemeinschaft (DFG), grant HO 3242/1–3. SP was supported by DFG, collaborate research area SFB 539-A4/C1.

References

Breiman, L.: Arcing classifiers (with discussion). The Annals of Statistics **26**, 801–849 (1998)

Breiman, L.: Prediction games and arcing algorithms. Neural Computation **11**, 1493–1517 (1999)

Breiman, L.: Random forests. Machine Learning **45**, 5–32 (2001)

Brier, G.: Verification of forecasts expressed in terms of probability. Monthly Weather Review **78**, 1–3 (1950)

Bühlmann, P.: Boosting for high-dimensional linear models. The Annals of Statistics **34**, 559–583 (2006)

Bühlmann, P., Hothorn, T.: Boosting algorithms: Regularization, prediction and model fitting (with discussion). Statistical Science **22**, 477–522 (2007)

Bühlmann, P., Yu, B.: Boosting with the L_2 loss: Regression and classification. Journal of the American Statistical Association **98**, 324–338 (2003)

Consul, P., Jain, G.: A generalization of the Poisson distribution. Technometrics **15**, 791–799 (1973)

Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association **97**, 77–87 (2002)

Efron, B., Johnston, I., Hastie, T., Tibshirani, R.: Least angle regression. The Annals of Statistics **32**, 407–499 (2004)

Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Association **96**, 1348–1360 (2001)

Fitzpatrick, T.B.: The validity and practicality of sun-reactive skin types I through VI. Archives of Dermatology **124**, 869–871 (1988)

Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**, 119–139 (1997)

Friedman, J.H.: Greedy function approximation: A gradient boosting machine. The Annals of Statistics **29**, 1189–1232 (2001)

Friedman, J.H., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting (with discussion). The Annals of Statistics **28**, 337–407 (2000)

Gallagher, R.P., McLean, D.I., Yang, C.P., Coldman, A.J., Silver, H.K., Spinelli, J.J., Beagrie, M.: Suntan, sunburn, and pigmentation factors and the frequency of acquired melanocytic nevi in children. Similarities to melanoma: The Vancouver mole study. Archives of Dermatology **126**, 770–776 (1990)

Gneiting, T., Raftery, A.E.: Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association **102**, 359–378 (2007)

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfeld, C.D., Lander, E.S.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science **286**, 531–537 (1999)

Hastie, T., Tibshirani, R.: Generalized Additive Models. Chapman & Hall, London (1990)

Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2 edn. Springer, New York (2009)

Hilbe, J.M.: Negative Binomial Regression. Cambridge University Press, Cambridge (2007)

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., Hofner, B.: mboost: Model-Based Boosting (2008). R package version 1.0-4. http://R-forge.R-project.org

Li, L.: Multiclass boosting with repartitioning. In: Proceedings of the 23rd International Conference on Machine Learning (ICML2006), pp. 569–576. New York: ACM Press (2006)

McCullagh, P., Nelder, J.A.: Generalized Linear Models. 2 edn. Chapman & Hall, London (1989)

Mullahy, J.: Specification and testing of some modified count data models. Journal of Econometrics **33**, 341–365 (1986)

Park, M.Y., Hastie, T.: L_1-regularization path algorithm for generalized linear models. Journal of the Royal Statistical Society, Series B **69**, 659–677 (2007)

Pfahlberg, A., Uter, W., Kraus, C., Wienecke, W.R., Reulbach, U., Kölmel, K.F., Gefeller, O.: Monitoring of nevus density in children as a method to detect shifts in melanoma risk in the population. Preventive Medicine **38**, 382–387 (2004)

R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008). URL http://www.R-project.org. ISBN 3-900051-07-0

Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning **37**, 297–336 (1999)

Schmid, M., Hothorn, T.: Boosting additive models using component-wise P-splines. Computational Statistics & Data Analysis **53**, 298–311 (2008a)

Schmid, M., Hothorn, T.: Flexible boosting of accelerated failure time models. BMC Bioinformatics **9:269** (2008b)

Segal, M.R.: Microarraygene expression data with linked survival phenotypes: Diffuse large-B-cell lymphoma revisited. Biostatistics **7**, 268–285 (2006)

Sun, Y., Todorovic, S., Li, J.: Unifying multi-class AdaBoost algorithms with binary base learners under the margin framework. Pattern Recognition Letters **28**, 631–643 (2007)

Tibshirani, R.: Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society, Series B **58**, 267–288 (1996)

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused Lasso. Journal of the Royal Statistical Society, Series B **67**, 91–108 (2005)

Uter, W., Pfahlberg, A., Kalina, B., Kölmel, K.F., Gefeller, O.: Inter-relation between variables determining constitutional UV sensitivity in Caucasian children. Photodermatology, Photoimmunology & Photomedicine **20**, 9–13 (2004)

Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B **68**, 49–67 (2006)

Zhu, J., Rosset, S., Zou, H., Hastie, T.: A multi-class AdaBoost. Technical Report 430, Department of Statistics, University of Michigan (2005)

Zou, H.: The adaptive Lasso and its oracle properties. Journal of the American Statistical Association **101**, 1418–1429 (2006)

Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B **67**, 301–320 (2005)