Advances in Boosting of Temporal and Spatial Models

Nikolay Robinzonov

München 2012

Advances in Boosting of Temporal and Spatial Models

Nikolay Robinzonov

Dissertation
zur Erlangung des Grades Doctor rerum naturalium an der
Fakultät Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

vorgelegt von
Nikolay Robinzonov

München, den 10. Oktober 2012

Erstgutachter:	Prof. Dr. Torsten Hothorn
Zweitgutachter:	Prof. Stefan Mittnik, PhD
Externer Gutachter:	Prof. Dr. Thomas Kneib

Datum der Einreichung:	10. Oktober 2012
Tag der mündlichen Prüfung:	30. Januar 2013

Abstract

Boosting is an iterative algorithm for functional approximation and numerical optimization which can be applied to solve statistical regression-type problems. By design, boosting can mimic the solutions of many conventional statistical models, such as the linear model, the generalized linear model, and the generalized additive model, but its strength is to enhance these models or even go beyond. It enjoys increasing attention since a) it is a generic algorithm, easily extensible to exciting new problems, and b) it can cope with "difficult" data where conventional statistical models fail. In this dissertation, we design autoregressive time series models based on boosting which capture nonlinearity in the mean and in the variance, and propose new models for multi-step forecasting of both.

We use a special version of boosting, called componentwise gradient boosting, which is innovative in the estimation of the conditional variance of asset returns by sorting out irrelevant (lagged) predictors. We propose a model which enables us not only to identify the factors which drive market volatility, but also to assess the specific nature of their impact. Therefore, we gain a deeper insight into the nature of the volatility processes. We analyze four broad asset classes, namely, stocks, commodities, bonds, and foreign exchange, and use a wide range of potential macro and financial drivers. The proposed model for volatility forecasting performs very favorably for stocks and commodities relative to the common GARCH(1,1) benchmark model. The advantages are particularly convincing for longer forecasting horizons. To our knowledge, the application of boosting to multi-step forecasting of either the mean or the variance has not been done before.

In a separate study, we focus on the conditional mean of German industrial production. With boosting, we improve the forecasting accuracy when compared to several competing models including the benchmark in this field, the linear autoregressive model. In an exhaustive simulation study we show that boosting of high-order nonlinear autoregressive time series can be very competitive in terms of goodness-of-fit when compared to alternative nonparametric models.

Finally, we apply boosting in a spatio-temporal context to data coming from outside the econometric field. We estimate the browsing pressure on young beech trees caused by the game species within the borders of the Bavarian Forest National Park "Bayerischer Wald," Germany. We found that using the geographic coordinates of the browsing cases contributes considerably to the fit. Furthermore, this bivariate geographic predictor is better suited for prediction if it allows for abrupt changes in the browsing pressure.

Zusammenfassung

Boosting ist ein iterativer Algorithmus zur Funktionsapproximation und numerischen Optimierung, der zur Lösung statistischer regressions-ähnlicher Probleme eingesetzt werden kann. Per Konstruktion weist Boosting große Ähnlichkeiten zu den Lösungen klassischer statistischer Modelle wie z.B. dem linearen Modell, dem generalisierten linearen Modell oder dem generalisierten additiven Modell auf, seine Stärke liegt jedoch in der Erweiterung dieser Modelle. Boosting erlangt zunehmende Aufmerksamkeit, da es a) ein generischer Algorithmus ist, der leicht auf neue spannende Problemstellungen erweitert werden kann und es b) mit "schwierigen" Daten umgehen kann, an denen herkömmliche statistische Modelle scheitern. Mittels Boosting entwickeln wir in dieser Dissertation autoregressive Zeitreihenmodelle, die Nichtlinearität im Mittelwert und in der Varianz erfassen, und schlagen neue Modelle für Mehrschritt-Prognosen vor.

Wir betrachten eine spezielle Version des Boosting, die "componentwise gradient boosting" genannt wird. Dieses Verfahren kann auf innovative Weise die für die bedingte Varianz irrelevanten (verzögerten) Prädiktoren aus dem Modell entfernen. Wir schlagen ein Modell vor, das Einflussfaktoren auf die Marktvolatilität identifiziert und schätzt. Dadurch bietet sich ein tiefer Einblick in die Form des Volatilitätsprozesses. Unter Verwendung einer breiten Auswahl von Makro- und Finanzfaktoren analysieren vier Anlageklassen: Aktien, Rohstoffe, Anleihen und Wechselkurse. Das vorgeschlagene Modell übertrifft das Benchmarkmodell GARCH(1,1) in der Prognose der Volatilität von Aktien und Rohstoffen. Die Überlegenheit ist für längerfristige Prognosen besonders deutlich. Nach unserem Kenntnisstand wird Boosting in dieser Arbeit erstmals auf Mehrschritt-Prognosen des Mittelwerts und der Varianz angewendet.

Ein weiterer Teil dieser Arbeit setzt den Schwerpunkt auf die Modellierung des bedingten Mittelwerts der deutschen Industrieproduktion. Mittels Boosting lässt sich die Vorhersagequalität im Vergleich zu mehreren Alternativmodellen verbessern, einschließlich dem linearen autoregressiven Modell, das als Benchmarkmodell in diesem Bereich dient. In einer umfassenden Simulationsstudie zeigen wir, dass Boosting nichlinearer Zeitreihen höherer Ordnung hinsichtlich der Anpassungsgüte sehr konkurrenzfähig gegenüber nichtparametrischen Modellen ist.

Schließlich wenden wir Boosting auf zeitlich-räumlich strukturierte Daten außerhalb des Ökonometriebereichs an. Wir schätzen die Intensität von Wildverbiss an jungen Buchen im Gebiet des Nationalparks "Bayerischer Wald", Deutschland. Ein Ergebnis dieser Studie ist, dass die Berücksichtigung der geographischen Koordi-

naten entscheidend zur Anpassungsgüte beiträgt. Weiter eignet sich der bivariate geographische Prädiktor besser zur Vorhersage, wenn er plötzliche Änderungen der Verbissintensität zulässt.

Acknowledgments

I share the credit of my work with many people who deserve sincere appreciation for their direct or indirect support. I am indebted to:

- Torsten Hothorn and Stefan Mittnik for their excellent guidance and remarkable influence on my research. Particularly, I am grateful to Torsten Hothorn for valuable insights into the world of boosting, to Stefan Mittnik for the right directions in the field of econometrics, and to both of them for encouraging me to combine these.
- Thomas Kneib for the external evaluation of this work.
- Matthias Schmid for insightful discussions about boosting.
- Martin Spindler for tireless co-authoring support and friendly help.
- Christian Groll for careful reading and critical notes.
- Esther Herberich for countless patient answers.
- All colleagues with whom I have collaborated in the essays of this thesis including Benjamin Hofner, Andreas Mayr, and Gerhard Tutz.
- My colleagues in the Department of Statistics at the University of Munich and especially to Andreas Fuest, Fabian Spanhel, Serkan Yener, and Tina Yener.
- My parents Teodora and Rumyan for their support, joy, worry, care, and unconditional love.
- My wife Milena and daughter Andrea to whom I dedicate this work...

  “Andrea, I wish you a good start in life.” ☺

Contents

Chapter 1

Introduction

Knowledge discovery in data concerns both the statistical and the machine learning communities. Often, the practical usefulness of this effort is twofold: uncovering an existing relationship between input and output variables (interpretation), and foretelling the output values conditioned on the observed input variables (prediction). Even though the communities disagree on how conclusions should be drawn from the data, see, e.g., Breiman (2001b) "Statistical Modeling: The Two Cultures," there exist algorithms which enjoy popularity in both fields. One such algorithm is boosting, and it is the methodological focus of this dissertation.

Boosting originated from the machine learning community (Freund and Schapire, 1996). Later, it was adopted in the statistical community by Friedman, Hastie, and Tibshirani (2000) and Friedman (2001) and is nowadays a versatile and realistic problem solving utility. It enjoys an ever increasing popularity since a) it is a generic algorithm, applicable in many situations, which addresses exciting new problems, and b) it can cope with "difficult" data where conventional statistical models fail. The latter is remarkable since correlated or ultra high-dimensional data situations are ill-posed problems for classical statistical estimation. Boosting is an algorithmic solution to such situations which does not sacrifice either of the two objectives mentioned above: interpretation and prediction.

Boosting was originally intended to solve two-class classification problems by maximizing the confidence of some predictive algorithm, in this case a binary classifier, but in a different context it can also be a regression. It suffices that the algorithm performs only slightly better than random guessing, in order to achieve an arbitrarily high accuracy. Therefore, it is called a *weak learner* or, as it is referred to in this dissertation, a *base learner*.

A base learner is typically, but not necessarily, a well-known regression-type statistical model, such as linear regression, GAM, or regression tree which models the connection between the response and the covariates. Boosting iteratively builds up the solution in small steps, where each step is based on the previous ones. This is done by repeatedly training the base learners on slightly changing versions of the original data until no signal remains in the data. In Chapter 2 we propose the formal definition of the generic algorithm and discuss several of its extensions.

In this dissertation, we use a special version of boosting called componentwise gradient boosting, which allows many base learners to individually specify the connection between the (groups of) covariates and the response. The algorithm repeatedly updates a small subgroup of the original base learner candidates in a series of iterations. Provided that the algorithm terminates reasonably soon, the variables which have been considered up to that termination point form an *active set* of variables. This implies an implicit exclusion of the remaining ones and, therefore, this version of boosting proposes a built-in component selection mechanism. Such data-driven decision on the relevance of variables is useful for model selection and is invaluable in the context of ultra high-dimensional data.

For the most part of this thesis, we apply boosting to time series with random output variables Y_t whose equally spaced observations are denoted by $y_t, t = 1, \dots, T$, where T is the sample size. As stated in Tsay (2005, p. 31) a purely stochastic time series Y_t is said to be linear if it can be represented as the moving average function of present and past error terms, also called innovations or shocks, ε_t

$$Y_t = \mu + \sum_{i=0}^{\infty} b_i \varepsilon_{t-i}, \tag{1.1}$$

where μ is a constant, $b_0 = 1$, $b_i, i > 1$ are real numbers, and ε_t are independent and identically distributed random variables from a continuous distribution with finite mean and variance. Provided $\mathbb{V}(\varepsilon_t) = \sigma_\varepsilon^2$ and $\sigma_\varepsilon^2 \sum_{i=1}^{\infty} b_i^2 < \infty$, then the first and the second moments of Y_t are time invariant, i.e., it is weakly stationary. Often, this infinite moving average can be represented by a low order autoregressive process or a combination of an autoregressive and a moving average part, but any stochastic process that does not satisfy (1.1) is said to be nonlinear (Tsay, 2005, p. 154). Assuming linearity in the time series models has often been found too restrictive since economic and financial systems naturally exhibit structural or behavioral changes. Therefore, modeling nonlinearity in financial and macroeconomic data is often desirable and is the practical focus of this dissertation.

The above definition of nonlinearity is too general to be of any practical use, so we focus on nonlinearity in the first and in the second conditional moments of Y_t. Let $\mathcal{F}_{t-1}$ denote the σ-field generated by the information available at some earlier time point $t-1$. This information is typically represented by a combination of the autoregressive elements Y_{t-i}, the innovations, and some exogenous variables. Then, the conditional mean and variance of Y_t given $\mathcal{F}_{t-1}$ are

$$\mu_t = \mathbb{E}\left(Y_t|\mathcal{F}_{t-1}\right) =: u(\mathcal{F}_{t-1}) \quad \text{and} \quad \sigma_t^2 = \mathbb{V}\left(Y_t|\mathcal{F}_{t-1}\right) =: v(\mathcal{F}_{t-1}), \tag{1.2}$$

respectively. If $u(\cdot)$ is nonlinear, Y_t is said to be nonlinear in mean. Boosting extensions for modeling such a kind of nonlinearity are proposed in Chapter 3. If $v(\cdot)$ is nonlinear, Y_t is said to be nonlinear in variance or heteroskedastic. In Chapter 4 we propose boosting techniques for modeling the dynamics of such hereroskedastic time series.

Most nonlinear time series models are concerned with the conditional mean in (1.2). These nonlinear models are divided into two groups: parametric and nonparametric. Nonlinear parametric models have one substantial drawback, which is the reason for their varying performance. They require an a priori choice of parametric functions, which are believed to be appropriate in certain situations. This approach is used mainly in financial applications, when we have sufficient knowledge to specify the nonlinear structure between the covariates and the response. Examples are models which assume different dynamics in different states of the world, or *regimes*, such as the threshold autoregressive (TAR) model (Tong, 1978), the smooth transition autoregressive (STAR) model[1] (Chan and Tong, 1986), or the Markov switching model (Hamilton, 1989). The bilinear model of Granger and Andersen (1978), which includes interactions between the times series and the innovations, is another parametric extension of Equation (1.1).

In contrast, the nonparametric time series models on which we focus are methods which estimate the mean nonlinearity in a data driven way. Examples are Friedman's (1991) multivariate adaptive regression spline (MARS) model applied to a time series context by Lewis and Stevens (1991), or the nonlinear additive autoregressive (NAAR) model proposed by Chen and Tsay (1993). Still, component selection, multicollinearity, or high-dimensionality of the input space plague these methods. By applying componentwise boosting to nonparametric autoregressive models with potentially many endogenous and exogenous lags (Chapter 3), we address these problems in a framework with minimal subjective requirements.

[1] Not to be confused with the structured additive regression (STAR) model (Fahrmeir, Kneib, and Lang, 2004).

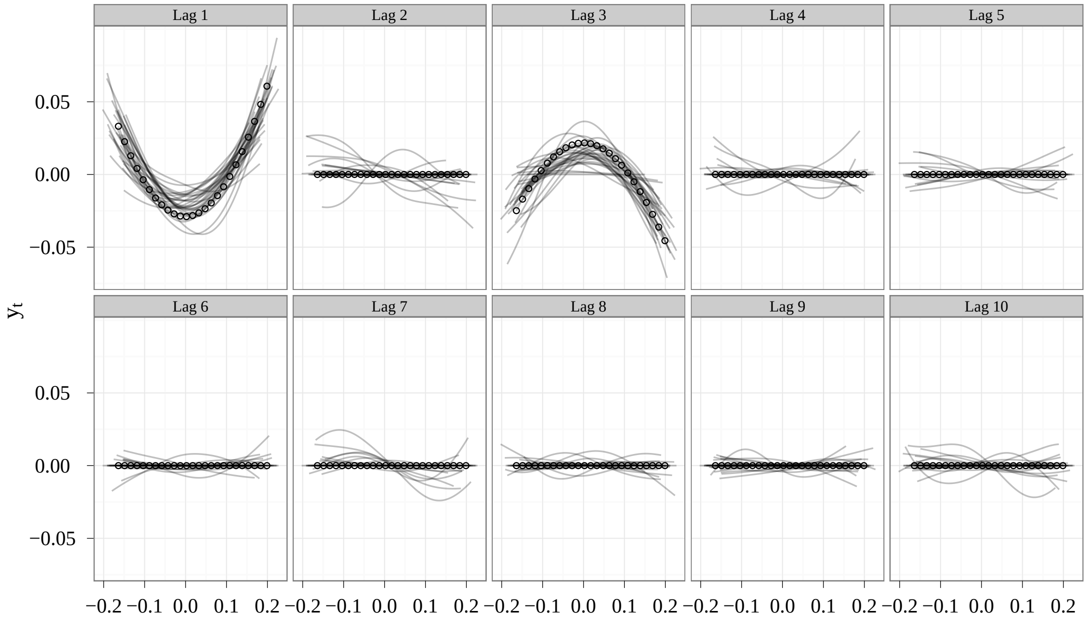


Figure 1.1: Estimating nonlinearity in the mean. In this example we show several estimations (solid lines) of a mean nonlinear autoregressive process given in Section 3.3, Table 3.1 (the NLAR2c model). The lag influence on the true mean dynamics are represented by the dotted lines, i.e., the relevant lags are one and three.

The simulation in Figure 1.1 shows, by example, the flexibility of our method. It depicts the estimation results of thirty simulations of a nonlinear autoregressive process. The exact definition of this process is deferred until Section 3.3, but for the illustrative purpose of this introduction we consider the relevant lags which are one and three, denoted by the circled lines. All other lags up to ten do not contribute to the mean dynamics, and are included in the model for checking robustness against false detection. The a priori information is the additive lag structure and neither the relevant lag nor their functional form are provided to the model. The proposed boosting method recovered the true underlying dynamics fairly closely, as shown by the solid lines estimated in a series of repetitions. Although not completely ignored, the redundant lags $2, 4$–10 were estimated close to zero. This resulted in an overall strong goodness-of-fit performance when compared to several competing methods.

Understanding the variance, or the volatility, seems to be an equally exciting topic as the mean. The interest in variance modeling started mostly with the seminal works of Engle (1982) and Bollerslev (1986) and has since become an intensely

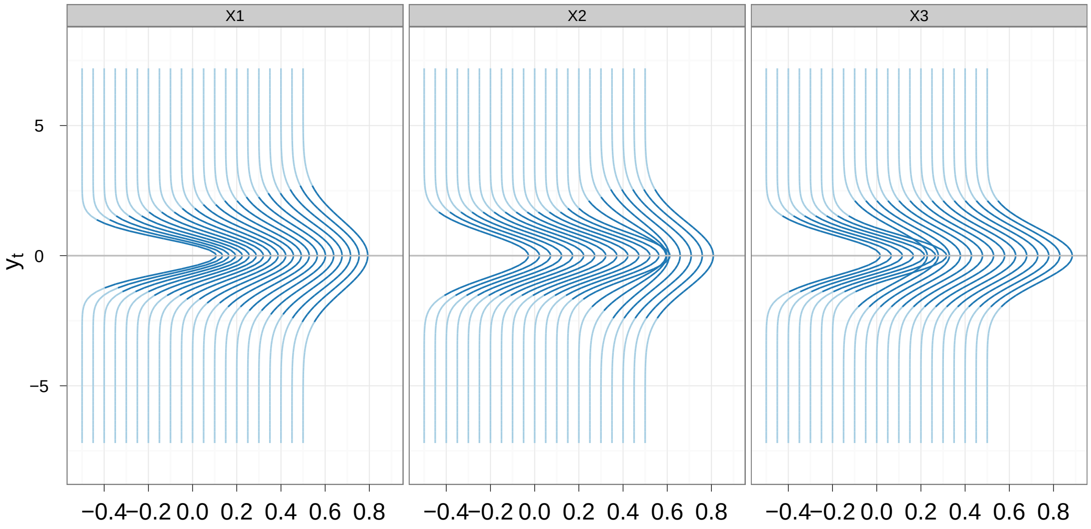


Figure 1.2: Estimating nonlinearity in the variance. An example in which the volatility of Y_t is driven by three factors. The dark lines indicate the estimated 95% interquantile range, the lighter ones show the estimated tails. The true data generating process is defined in Chapter 4, Equation 4.6.

researched field in financial econometrics since it is our tool for measuring risk. The importance of understanding and adequately modeling financial market risk is widely recognized and has again become evident during recent turbulences in the markets. Volatility forecasts are used for risk management purposes, for example, to project risk measures, such as Value at Risk (VaR) and Expected Shortfall (ES), or to decide on hedging or other risk mitigation strategies. They are also used for dynamic asset allocation decisions that are not just based on asset specific risk but also on the dependence between assets, expressed in terms of time varying, volatility dependent measures, such as correlations or betas.

Using the well established GAM framework, as in the NAAR model for example, implies that the conditional distribution of the response belongs to the exponential family. Assuming an exponential family we generally have the advantage of flexibly modeling the conditional mean but we “sacrifice” the role of the conditional variance since exponential families rarely allow modeling of parameters other than the mean. It is, however, possible to obtain general expressions for the mean and for the variance of exponential family distributions, in terms of their dispersion (or scale) parameter and their family specific functions. But at some point we are confronted with the limitations of the exponential family assumption.

Therefore, we used componentwise boosting, which is tailor made for estimating the conditional variance of asset returns and sorting out irrelevant (lagged) predictors. We propose a model which answers whether and, if so, how, macro factors influence the volatility of asset prices. By boosting, we gain deeper insight into the nature of the volatility processes. As will be shown, boosting techniques enable us not only to identify the factors driving market volatility, but also to assess the specific nature of their impact (see for example Figure 1.2) and, ultimately, help to improve prediction. Employing a broad set of potential macroeconomic and financial variables, we specify a flexible model that is capable of capturing their linear and nonlinear influences on volatility.

Finally, we apply boosting in a spatio-temporal context. The focus in Chapter 5 is the estimation of the conditional browsing probabilities in the Bavarian Forest National Park "Bayerischer Wald," Germany. Forest regeneration is hindered at a very early stage by the browsing damage caused by various game species. In middle Europe, especially, roe and red deer are the most common species browsing on young trees. The consequences of excessive browsing often lead to forest growth retardation and homogenization. Developing precise measures to reflect the true condition of the forest's regeneration is thus crucial and nontrivial.

In summary, this dissertation is organized as follows:

Chapter 2. In this section we explain the underlying idea and give a detailed insight into the technicalities of boosting. We further consider several topics (personally preferred but not entirely arbitrarily selected) which emphasize existing problems of boosting. We also comment on several improvements of the boosting algorithm and on existing connections to other related methods. Section 2.4 is based in part on Hofner, Mayr, Robinzonov, and Schmid (2012), "Model-based boosting in R: A hands-on tutorial using the R package *mboost*," *Computational Statistics*, 1–33.

Chapter 3. By letting the covariates be lagged values of a time series, we apply boosting to identify the relevant lags and forecast the conditional mean. An exhaustive simulation study shows that boosting high-order autoregressive time series can be very competitive in terms of dynamics estimation. Furthermore, we conduct a forecasting comparison over the monthly growth rates of German industrial production. The inclusion of different exogenous variables (leading indicators) improved the forecasting performance. Chapter 3 is based on Robinzonov, Tutz, and Hothorn (2012), “Boosting techniques for nonlinear time series models,” *AStA Advances in Statistical Analysis 96*, 99–122.

Chapter 4. Using monthly data, we rely on boosting techniques based on regression trees as base learners to identify relevant volatility drivers as well as the functional form of their influence. We analyze the determinants of volatility in the four broad asset classes of stocks, commodities, bonds, and foreign exchange, making use of a wide range of potential macro and financial drivers. Using realized volatility as a proxy for the unobserved volatility we conduct an out-of-sample forecasting study in which we show that boosting performs very well for stocks and commodities relative to the common GARCH(1,1) benchmark model. Chapter 4 is based on Mittnik, Robinzonov, and Spindler (2012), “Boosting the Anatomy of Volatility,” http://epub.ub.uni-muenchen.de/12976/.

Chapter 5. We evaluate and compare several boosting models on binary data in a spatio-temporal context. The objective is to estimate a surface representing the browsing probabilities on young beech trees within the borders of the Bavarian Forest National Park “Bayerischer Wald” in southern Germany. In our model selection procedure, we found that the spatial component and the height of the trees do contribute considerably to the goodness-of-fit. Furthermore, we found that a spatial component which allows for abrupt changes in the browsing pressure is better suited for prediction than the smooth bivariate P-spline tensor product alternative. This is mostly due to the irregular distribution of the tree regeneration areas. Chapter 5 is based on Robinzonov and Hothorn (2010), “Boosting for estimating spatially structured additive models,” in *Statistical Modelling and Regression Structures. Festschrift in Honour of Ludwig Fahrmeir*, edited by Kneib and Tutz, pp. 181–196.

Chapter 6 This chapter gives our conclusions.

Chapter 2

Gradient Boosting

Boosting, in its original form as proposed by Freund and Schapire (1996), was intended to solve two-class classification problems by maximizing the confidence, or the “margins,” of a binary classificator. The AdaBoost algorithm, as it is called, is nowadays the most well known boosting algorithm. Excellent explanations of its algorithmic details can be found in Friedman et al. (2000); Bühlmann and Hothorn (2007a); Hastie, Tibshirani, and Friedman (2009a) and Bühlmann and van De Geer (2011) among others.

In summary, the purpose of AdaBoost is to enhance the predictive accuracy of some, already existing, classification algorithm, e.g., classification tree (Breiman, Friedman, Olshen, and Stone, 1984). This classification algorithm will be called a *weak learner* because it suffices that it performs only slightly better than random guessing in order to attain arbitrarily good accuracy (Kearns and Valiant, 1994; Schapire, Freund, Bartlett, and Lee, 1998). AdaBoost repeatedly applies the weak learner on successively changing versions of the original data. The changes are intended to re-weight the observed data in a way that misclassified observations receive more attention in the next iteration. To do this, the algorithm increases their weights in dependence on the training error, while the weights of the unproblematic ones are decreased. Therefore, the “hard cases” receive more attention by the weak learner and we iterate this M times. The number of iterations is the main tuning parameter for boosting and we will comment on this in Section 2.5, but for now we assume that we know the optimal M. Finally, we get an ensemble of M predictions which are suitably aggregated so that more accurate predictions have a larger contribution.

Combining rules to form an ensemble and to aggregate it in a final decision lies at the heart of ensemble learning techniques. Boosting is a special kind of sequential-ensemble learning. This can be very advantageous since boosting preserves interpretation—a property which does not apply to other parallel-ensemble learning techniques, such as bagging or random forests that yield "black box" predictions.

Boosting was placed in a regression framework by Friedman (2001) who explained it as a functional gradient descent (FGD) technique. This interpretation of boosting is also shared by Breiman (1998, 1999); Mason, Baxter, Bartlett, and Frean (2000); Bühlmann and Yu (2003); Rosset, Zhu, and Hastie (2004); Bühlmann and Hothorn (2007a) and many others. In these references, boosting is interpreted as a function optimization approach strikingly similar to the well known steepest descent optimization and we adhere to that interpretation in this thesis. Note, however, that any estimation process which recursively improves its predictive accuracy in small steps fits into the framework of boosting. It has, therefore, many variants, e.g., likelihood based boosting (Tutz and Binder, 2006) or forward stagewise additive modeling (Hastie et al., 2009a, Chapter 10). Likelihood based boosting has a different update and component selection mechanism, while forward stagewise additive modeling uses a slightly different way of regularization.

2.1 The Objective of Boosting

The objective of boosting is to estimate a function η that links a random outcome Y_t (or response) to an r dimensional random vector of covariates Z_t by minimizing the expectation of a loss function L, such that

$$\hat{\eta} = \underset{\eta}{\arg\min}\, \mathbb{E}\ [L(Y_t, \eta(\mathbf{z}_t))] \tag{2.1}$$

where $Y_t|Z_t = \mathbf{z}_t$ are considered independent and, in time series analysis, Z_t usually includes lagged values of Y_t in addition to other, previously observed, exogenous variables and eventually some deterministic covariates. Once observed, it makes no difference for the methodology whether the covariates in Z_t are random or deterministic. Its observations are denoted by $\mathbf{z}_t = (z_{t,1}, z_{t,2}, \ldots, z_{t,r})^\top \in \mathbb{R}^r$ and are observed backwards in time. For example, if we consider two lagged response values $(y_{t-1}, y_{t-2})^\top$ and two additional exogenous variables which are lagged only once $(x^{(1)}_{t-1}, x^{(2)}_{t-1})$, we have $r = 4$ and $\mathbf{z}_t = (z_{t,1}, \ldots, z_{t,4})^\top = (y_{t-1}, y_{t-2}, x^{(1)}_{t-1}, x^{(2)}_{t-1})^\top$. The exact structure of $\mathbf{z}_t$, i.e., the lag length, the number of exogenous variables, the inclusion of seasonal components etc., is very general and depends on the research question and the available data. Therefore, it is specified in the relevant places of the remaining chapters. Without loss of generality, we assume in this section that the intercept is zero. The regularization techniques which we discuss later usually do not penalize the intercept term and this assumptions is also made for simplicity. In addition, L is assumed to be differentiable and convex with respect to η.

Seeking for a solution in function space, (2.1) is flexible and certainly advantageous for prediction but too general and especially difficult of interpretation. This suggests that we often need a more specific and admittedly more restrictive solution by introducing some structure in η through parameters $\boldsymbol{\beta}$. We substitute the original function space solution for a parameter space solution in the following way,

$$\hat{\eta} = \arg\min_{\eta} \mathbb{E}\left[L(Y_t, \eta(\mathbf{z}_t; \boldsymbol{\beta}))\right] \tag{2.2}$$

where $\boldsymbol{\beta}$ is a finite- or infinite-dimensional parameter. Therefore, gradient boosting is applicable, but not restricted to, generalized linear models (GLM, McCullagh and Nelder, 1989). In this context, a finite-dimensional $\boldsymbol{\beta} = (\beta_1, \dots, \beta_r)^\top$ is represented by

$$\eta_t = \eta(\mathbf{z}_t; \boldsymbol{\beta}) = \beta_1 z_{t,1} + \cdots + \beta_r z_{t,r}$$

or in generalized additive models (Hastie and Tibshirani, 1990)

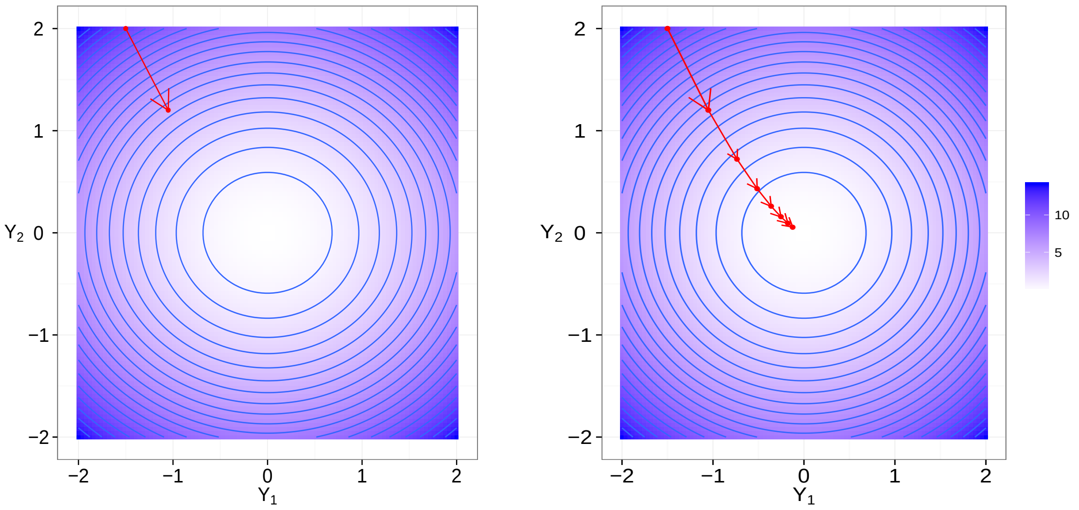
$$\eta_t = f_1(z_{t,1}; \beta_1) + \cdots + f_r(z_{t,r}; \beta_r)$$

with $k(\eta_t) = \xi(Y_t|Z_t = \mathbf{z}_t)$, k being a specified response function,[1] ξ is any desired characteristic of the conditional density, e.g., $\xi(Y_t|Z_t = \mathbf{z}_t) = \mathbb{E}\,(Y_t|Z_t = \mathbf{z}_t)$, $f_j, j = 1 \dots, r$ are unknown functions, and $\beta_j, j = 1, \dots, r$ are vectors. The exponential family—which is a central assumption in GLM—is by no means mandatory for gradient boosting and one can even estimate functionals of distribution-free conditional densities similar to median regression by suitably specifying the loss function L (see Section 2.4). The infinite-dimensional case of $\boldsymbol{\beta}$ is imaginable for regression trees with β_j describing the split points, and the constant values assigned to the response.

In practice, the solution must be found in the space spanned by the data. By the law of large numbers, the average of the loss function (the empirical loss or the empirical risk) converges to the expectation as T increases, i.e., $\frac{1}{T}\sum_{t=1}^{T} L(Y_t, \eta(\mathbf{z}_t; \boldsymbol{\beta})) \xrightarrow{\mathbb{P}} \mathbb{E}\left[L(Y_t, \eta(\mathbf{z}_t))\right]$, where $\xrightarrow{\mathbb{P}}$ denotes convergence in probability. Therefore, we approximate the theoretical and unknown expectation in (2.2) by

$$\hat{\eta} = \arg\min_{\eta} \frac{1}{T}\sum_{t=1}^{T} L(y_t, \eta(\mathbf{z}_t; \boldsymbol{\beta})), \tag{2.3}$$

[1] The response function is usually denoted by h, rather than by k. Since h indicates the base learner procedure (see below) we simply avoid ambiguity at this point.



(a) (b)

Figure 2.1: Steepest descent

where y_t denotes the observation of Y_t. The solution of (2.3) is found by successively reducing the empirical loss in a steepest-descent like algorithm. A careful specification of the loss function, L, leads to an estimation of the desired characteristic of the conditional distribution and is the main reason for boosting's versatility. We will consider several specifications of the loss function in the following sections.

2.2 Steepest Descent

As a short review of the pure steepest descent algorithm, let us consider the following example. Suppose we are to minimize the function $L(y_1, y_2) = 1.5y_1^2 + 2y_2^2$. The negative gradient is

$$g(y_1, y_2)^\top = \left(-\frac{\partial L}{\partial y_1}, -\frac{\partial L}{\partial y_2}\right) = (-3y_1, -4y_2).$$

This gradient shows the direction which reduces L most. For the steepest descent algorithm, we arbitrarily choose a starting point, e.g., $y^{[0]} = (-1.5, 2)$, and update the new position through $y^{[1]} = y^{[0]} + 0.1 \cdot g(y^{[0]}) = (-1.05, 1.2)$. This initial step is shown in Figure 2.1(a). The step length was chosen small, in this case 0.1, and we iterate $y^{[k]} = y^{[k-1]} + 0.1 \cdot g(y^{[k-1]})$ until the changes are small enough, Figure 2.1(b).

Similarly, boosting iteratively builds up the solution in small steps, where each step is based on the previous one. It favors the direction that reduces the empirical loss most, i.e., the direction specified by the negative gradient. Since we seek a solution in the data space, we repeatedly fit the covariates against the negative gradient. It is in essence the steepest descent optimization technique of a T-dimensional function with one major difference: the negative gradient is estimated. Since the dimensionality of the gradient depends on the sample size T, it can be regarded as an infinite-dimensional function. Therefore, we do not use the gradient itself, but rather take the direction of the greatest current correlation so that when new observations arrive, we would improve in terms of out-of-sample predictive accuracy.

Recall that the objective is to minimize (2.3). Given any current estimate $\hat{\boldsymbol{\beta}}^{[m-1]}$ (or $\eta(\mathbf{z}_t; \hat{\boldsymbol{\beta}}^{[m-1]})$), we compute the negative gradient,

$$g_t^{[m]} := -\left[\frac{\partial}{\partial \eta} L(y_t, \eta)\right]_{\eta=\eta(\mathbf{z}_t;\hat{\boldsymbol{\beta}}^{[m-1]})}, t = 1, \ldots, T, \tag{2.4}$$

which, similarly to the red line in Figure 2.1(a), gives the direction of steepest descent. Then, we estimate that gradient by some statistical model h,

$$\hat{\boldsymbol{\gamma}}^{[m]} = \arg\min_{\boldsymbol{\gamma}} \left[h(\mathbf{z}_t; \boldsymbol{\gamma}) \to \hat{g}_t^{[m]}\right], \tag{2.5}$$

and update η,

$$\eta(\mathbf{z}_t; \hat{\boldsymbol{\beta}}^{[m]}) = \eta(\mathbf{z}_t; \hat{\boldsymbol{\beta}}^{[m-1]}) + \nu \cdot \underbrace{h(\mathbf{z}_t; \hat{\boldsymbol{\gamma}}^{[m]})}_{\substack{\text{estimated} \\ \text{gradient} \\ \hat{g}_t^{[m]}}}, \quad t = 1, \ldots, T. \tag{2.6}$$

Note that the statistical model (or smoother) h is the base learner and ν is the step size or shrinkage parameter (Bühlmann and Hothorn, 2007a). A small shrinkage parameter, typically $\nu \approx 0.1$, can be interpreted as a local regularization parameter since the current improvement is only a small step away from the previous one. In this way we "cure" the typical instability of forward selection methods (Breiman, 1996). Forward selection methods are called "greedy" since they aim at maximal improvement of the objective function with each step, regardless of the impact on model complexity (Zhao and Yu, 2007). In Section 2.5, we discuss this aspect in more detail.

The base learner is typically, but not necessarily, a wellknown regression type statistical model, such as linear regression, GAM, or regression tree, which models the connection between the response and the covariates. This, in addition to the loss function specification, is the key for the versatility of boosting. The combinations of base learners and loss functions are diverse and wide ranging and motivate plenty of new model definitions. But behind the scenes, we have the same generic algorithm—the iteration of (2.4)–(2.6)—which is called boosting (Friedman, 2001).

If we rewrite $\eta(\mathbf{z}_t; \hat{\boldsymbol{\beta}}^{[m]})$ and the gradient $\hat{g}_t^{[m]}$ in vector notation, i.e., $\hat{\boldsymbol{\eta}}^{[m]} = \left[\eta(\mathbf{z}_1; \hat{\boldsymbol{\beta}}^{[m]}), \ldots, \eta(\mathbf{z}_T; \hat{\boldsymbol{\beta}}^{[m]})\right]^\top$ and $\hat{\boldsymbol{g}}^{[m]} = \left[h(\mathbf{z}_1; \hat{\boldsymbol{\gamma}}^{[m]}), \ldots, h(\mathbf{z}_T; \hat{\boldsymbol{\gamma}}^{[m]})\right]^\top$, we can represent (2.6) as an additive sum of the form

$$\hat{\boldsymbol{\eta}}^{[M]} = \sum_{m=1}^{M} \nu \hat{\boldsymbol{g}}^{[m]}, \tag{2.7}$$

where M is an optimal stop number of iterations. In addition, we have the same additive structure in the parameters

$$\hat{\boldsymbol{\beta}}^{[M]} = \sum_{m=1}^{M} \nu \hat{\boldsymbol{\gamma}}^{[m]}. \tag{2.8}$$

Therefore, the final parameter estimates can be expressed as an additive sum of the former estimates. This recursive aggregation of the parameter estimates explains the motivation behind referring to boosting as a forward stagewise additive technique (Hastie et al., 2009a). The additive structure here should not be confused with the similarly labeled additive structure of the basis expansion in GAM. Note also that Hastie et al. (2009a) do not use a fixed step-size ν but instead estimate an optimal step size $\nu^{[m]}$ for each iteration.[2]

The parameter M is regarded as the primary tuning parameter which controls the bias–variance tradeoff. It is usually chosen via some cross-validating assessment aiming for optimal out-of-sample prediction. Below we discuss several strategies for estimating M. In summary, the boosting algorithm is as follows:

1. Initialize the vector estimate $\hat{\boldsymbol{\eta}}^{[0]}$, e.g., for $L = \mathrm{L}_2$ (defined in Section 2.4), $\hat{\boldsymbol{\eta}}^{[0]} = \bar{y} \cdot (1, \ldots, 1)^\top$ with $\bar{y} = \frac{1}{n} \sum_{t=1}^{T} y_t$.
2. Set $m = 0$.
3. Increase m by one.

[2] In their original notation, the step size is denoted by α_m.

4. (a) Evaluate $\hat{\boldsymbol{\eta}}^{[m-1]}$ and compute the negative gradient (2.4).
   (b) Estimate the negative gradient as in Equation (2.5) which yields $\hat{\boldsymbol{g}}^{[m]}$.
   (c) Update $\hat{\boldsymbol{\eta}}^{[m]} = \hat{\boldsymbol{\eta}}^{[m-1]} + \nu \cdot \hat{\boldsymbol{g}}^{[m]}$, Equation (2.6).
5. Iterate Steps 3 and 4 until a final step M determined by some stopping condition.

2.3 Componentwise Boosting

The algorithm can be further refined by tweaking step (2.5). We make an individual model choice for each covariate–response pair[3] in a way which is believed to best describe their relationship. These r individual models represent r base learners, or equivalently weak learners, and we can think of them as r isolated subsolutions of the original optimization problem. This is called *componentwise* gradient boosting (Bühlmann and Yu, 2003). Instead of fitting all covariates at once, they are fitted separately against the gradient. At each boosting step, only one covariate is included, namely the one which most correlates with the negative gradient. This covariate is gradually updated until some other covariate gets more correlated in magnitude with the gradient. The new covariate is in turn smoothly blended into the model, and so on. This is another way of keeping the learner “weak” by simply restraining a complex structure with many parameters. The relationship between the covariates and the response is, as in regression modeling, an expert decision.

During the iterations, we repeatedly update a small subgroup of the original base learner candidates. Provided the algorithm terminates reasonably soon, the active set of variables, i.e., those with nonzero parameters, implies an implicit exclusion of the remaining ones. This is called *early stopping* and the result is a built in variable selection and model choice (Bühlmann and Yu, 2003; Kneib, Hothorn, and Tutz, 2009).

To formalize the component selection process, we assume the following structure $\eta_t(\mathbf{z}_t) = \sum_{j=1}^{r} f_j(z_{t,j})$ where r is the number of (groups of) covariates. The additional

[3] We do not necessarily have to isolate each single covariate since we could also group them and pair the groups with the response.

selection step modifies (2.5) in the following way:

$$\hat{\gamma}_j^{[m]} = \underset{\gamma_j}{\arg\min} \left[f_j(z_{t,j}; \gamma_j), \rightarrow \hat{g}_t^{[m]} \right], \quad j = \{1, \ldots, r\} \tag{2.5 a}$$

$$\hat{s}_m = \underset{j \in \{1,\ldots,r\}}{\arg\min} \sum_{t=1}^{T} \left(g_t^{[m]} - f_j(z_{t,j}; \hat{\gamma}_j^{[m]}) \right)^2 . \tag{2.5 b}$$

The modification implies that we first optimize with respect to the individual base learner parameters which results in r estimations $\hat{\gamma}_1^{[m]}, \ldots, \hat{\gamma}_r^{[m]}$. Then, according to the sum of the squared residuals criterion, we optimize with respect to the index j, i.e., $\hat{s}_m \in \{1, \ldots, r\}$, and the update vector is $\hat{\boldsymbol{\gamma}}^{[m]} = (0, \ldots, 0, \hat{\gamma}_{\hat{s}_m}^{[m]}, 0, \ldots, 0)^\top$ with zeros for all but the $\hat{s}_m$th component and

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^{[m]} &= \hat{\boldsymbol{\beta}}^{[m-1]} + \nu \hat{\boldsymbol{\gamma}}^{[m]} \\
&= \left(\beta_1^{[m-1]}, \ldots, \beta_r^{[m-1]} \right)^\top + \nu \cdot (0, \ldots, 0, \hat{\gamma}_{\hat{s}_m}^{[m]}, 0, \ldots, 0)^\top \\
&= \nu \cdot \left(\sum_{i=1}^{m-1} \gamma_1^{[i]}, \ldots, \sum_{i=1}^{m} \gamma_{\hat{s}_m}^{[i]}, \ldots, \sum_{i=1}^{m-1} \gamma_r^{[i]} \right)^\top .
\end{aligned} \tag{2.9}$$

Therefore, the update (2.6) becomes

$$\eta(\mathbf{z}_t; \hat{\boldsymbol{\beta}}^{[m]}) = \eta(\mathbf{z}_t; \hat{\boldsymbol{\beta}}^{[m-1]}) + \nu \ \cdot f_{\hat{s}_m}(\mathbf{z}_t; \hat{\boldsymbol{\gamma}}^{[m]}), \quad t = 1, \ldots, T. \tag{2.10}$$

We can combine different base learners for different variables. These individual specifications offer great flexibility. Furthermore, due to the additive update in (2.9), the estimate of a function f_j at iteration m has the same structure as the corresponding base learner. Depending on the circumstances, we could easily combine established statistical models into one, and the structural assumption of the model will be specified by the base learners. Many base learners can be represented as simple penalized least squares models with a general notation of the form

$$\hat{\boldsymbol{g}}^{[m]} = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{K})^{-1} \mathbf{X}^\top \boldsymbol{g}^{[m]} = \mathcal{S} \boldsymbol{g}^{[m]}, \tag{2.11}$$

where the hat-matrix is defined by $\mathcal{S} = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{K})^{-1} \mathbf{X}^\top$, with design matrix $\mathbf{X}$, penalty parameter λ, and penalty matrix $\mathbf{K}$ (the index j is omitted for convenience). The design and penalty matrices depend on the type of the base learner and this notation allows for linear, categorical, smooth, or even spatial effects (see Hofner, Hothorn, Kneib, and Schmid, 2011; Hofner et al., 2012, for details). Specifically, given that the covariate is continuous, (2.11) represents a P-spline estimation (Eilers and Marx, 1996), otherwise, if it is discrete, (2.11) represents a ridge regression (see Hoerl and Kennard (1970) for nominal and Gertheiss and Tutz (2009) for ordinal categories). A penalty parameter $\lambda = 0$ results in unpenalized estimation. Other learners such as regression trees, considered in Chapters 4 and 5, are also possible. In summary, the componentwise boosting algorithm is the following:

1. Initialize the vector estimate $\hat{\boldsymbol{\eta}}^{[0]}$, e.g., for $L = \mathrm{L}_2$, $\hat{\boldsymbol{\eta}}^{[0]} = \bar{y} \cdot (1, \ldots, 1)^\top$ with $\bar{y} = \frac{1}{n} \sum_{t=1}^{T} y_t$.

2. Set $m = 0$. Specify the set of base learners and denote the number of base learners by r.

3. Increase m by one.

4. (a) Evaluate $\hat{\boldsymbol{\eta}}^{[m-1]}$ and compute the negative gradient (2.4).

   (b) Estimate the negative gradient as in Equation (2.5 a) which yields r vector estimations, i.e., $\hat{\boldsymbol{g}}_1^{[m]}, \ldots, \hat{\boldsymbol{g}}_r^{[m]}$.

   (c) Select the base learner $\hat{s}_m \in \{1, \ldots, r\}$, Equation (2.5 b), that most correlates with the gradient according to the residual-sum-of-squares criterion. Therefore, $\hat{\boldsymbol{g}}_{\hat{s}_m}^{[m]} = f_{\hat{s}_m}(\mathbf{z}_t; \hat{\boldsymbol{\gamma}}^{[m]})$ is the selected estimate of the gradient vector.

   (d) Update $\hat{\boldsymbol{\eta}}^{[m]} = \hat{\boldsymbol{\eta}}^{[m-1]} + \nu \cdot \hat{\boldsymbol{g}}_{\hat{s}_m}^{[m]}$, Equation (2.10).

5. Iterate Steps 3 and 4 until the final step, as determined by the stopping condition.

2.4 Loss Functions

Depending on the specification of the loss function, we can estimate any desired characteristic of the conditional distribution of the response. This, coupled with the large number of base learners, guarantees a rich set of models that can be fitted using boosting. We can specify the connection between the response and the covariates in a fairly modular way, such as

$$\xi(y|\mathbf{z}) = \hat{f}_1 + \cdots + \hat{f}_r, \tag{2.12}$$

having on the right hand side any desired combination of base learners. On the left hand side, $\xi(\cdot)$ describes *some* characteristic of the conditional distribution. In the following subsections, we discuss the major aspects related to the choice of the family.

2.4.1 Continuous Response

In the case of a Gaussian continuous response, our assumption is that $Y|Z$ is normally distributed and the loss function is the negative Gaussian log-likelihood, which is equivalent to the L_2 loss

$$L(y, \eta) = \frac{1}{2}(y - \eta)^2$$

(see Figure 2.2(a)). The corresponding negative gradient is simply $(y - \eta)$, which turns out to be the residuals vector.

By boosting, we can also implement a distribution-free, median regression approach especially useful for long-tailed error distributions. In this case, we use the L_1 loss defined as

$$L(y, f) = |y - \eta|$$

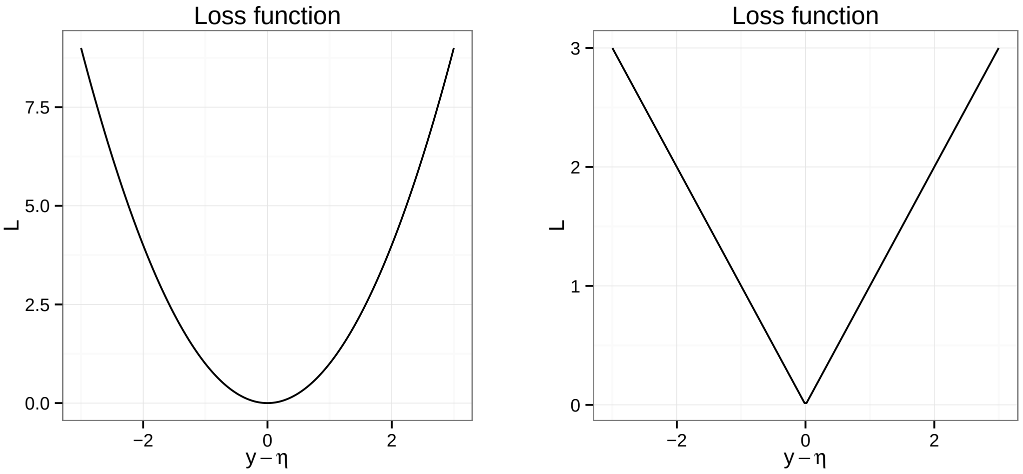
and shown in Figure 2.2(b) which means that we are interested in the median of the conditional distribution. Note that the L_1 loss is not differentiable at $y = \eta$ and the value of the negative gradient at such points is fixed at zero. Since the probability of a real-valued random variable to result in exactly zero is zero, this means that this is neither a theoretical nor an empirical issue.

A compromise between the L_1 and the L_2 loss is the Huber loss function shown in Figure 2.3(a). It is defined as

$$L(y, \eta; \delta) = \begin{cases} (y - \eta)^2/2 & \text{if } |y - \eta| \leq \delta, \\ \delta(|y - \eta| - \delta/2) & \text{if } |y - \eta| > \delta \end{cases}$$

where the parameter δ limits the outliers which are subject to absolute error loss. The Huber loss can be seen as a robust alternative to the L_2 loss. One can either specify δ subjectively, e.g., $\delta = 2$, or leave it to be adaptively chosen by the boosting algorithm. An adaptive specification of δ, proposed by Friedman (2001), means that each boosting step produces a new $\delta^{[m]}$ matching the actual median of the absolute values of the residuals, i.e.,

$$\delta^{[m]} = \text{median}\left(\left|y_i - \hat{\eta}^{[m-1]}(x_i)\right|, i = 1, \ldots, n\right).$$



(a) L_2 loss function

(b) L_1 loss function

Figure 2.2: The loss function allows flexible specification of the link between the response and the covariates. The figure on the left hand side illustrates the L_2 loss, the figure on the right hand side shows the L_1 loss function.

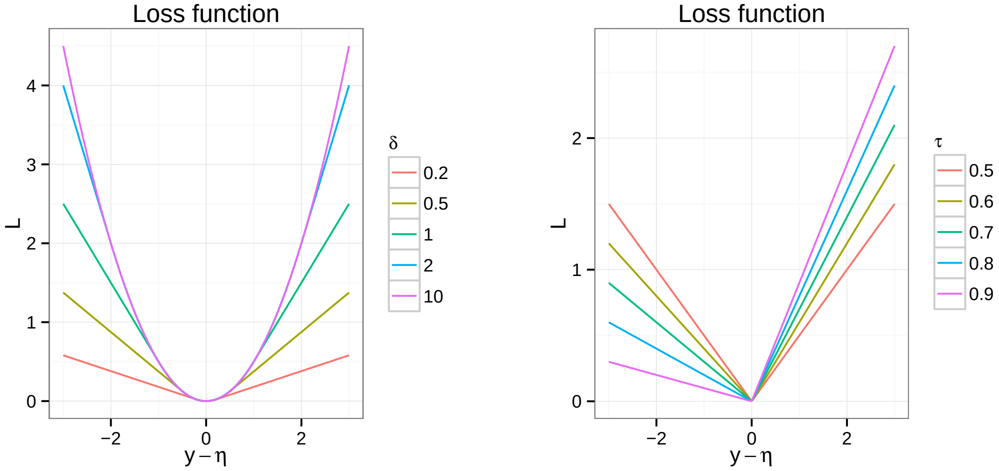
Another alternative for settings with continuous response is modeling the conditional quantiles through quantile regression (Koenker, 2005; Fenske, Kneib, and Hothorn, 2011). The main advantage of quantile regression is (beyond its robustness towards outliers) that it does not rely on any distributional assumptions on the response or the error terms. The appropriate loss function here is the check function shown in Figure 2.3(b). The special case of the 0.5 quantile leads to median regression.

If we are interested in the conditional variance, we can model it through the exponential link function, i.e., $k(\eta_t) = \exp(\eta(\mathbf{z}_t)) = \mathbb{V}\left(Y_t | Z_t = \mathbf{z}_t\right)$. Assuming that $Y_t | Z_t \sim N(0, e^{\eta_t})$ is reasonable, the negative conditional log-likelihood function is used as loss the function and the result is

$$L_t = \frac{1}{2}\left[\eta_t + \frac{y_t^2}{e^{\eta_t}}\right] \tag{2.13}$$

(after some simplifications) with the corresponding negative gradient given by

$$g_t = -\frac{\partial L_t}{\partial \eta_t} = \frac{1}{2}\left[\frac{y_t^2}{e^{\eta_t}} - 1\right]. \tag{2.14}$$



(a) Huber loss function (b) Quantile regression

Figure 2.3: The Huber loss function on the left hand side is useful when robustness is a concern. It adaptively changes the limit for L_1 penalization of outliers. The figure on the right hand side illustrates several examples of the check function loss with different quantiles ($\tau = 0.5$ is the default).

See Chapter 4, which is based on Mittnik, Robinzonov, and Spindler (2012), for further details.

2.4.2 Binary Response

Analogously to Gaussian regression, the probability parameter of a binary response can be estimated by minimizing the negative binomial log-likelihood

$$\begin{aligned} L(y, \eta) &= -\left[y\ \log\big(\pi(\eta)\big) + (1-y)\ \log\big(1-\pi(\eta)\big)\right] \\ &= \log(1 + \exp(-2\,\tilde{y}\eta)) \end{aligned} \tag{2.15}$$

where $\tilde{y} = 2y-1$ and $\pi(\eta) = \mathbb{P}(Y = 1|\mathbf{z})$. In Equation (2.15), the $\tilde{y}\eta$ are the so-called margin values (depicted in Figure 2.4) which are, roughly speaking, the equivalent of the continuous residuals $y - \eta$ for the binomial case. This recoding means that the negative binomial log-likelihood loss and the exponential loss (defined below) coincide in their population minimizer (see Bühlmann and Hothorn, 2007a, Section 3). For further details or examples of boosting with the negative binomial log-likelihood loss function, see Chapter 5 or Robinzonov and Hothorn (2010).

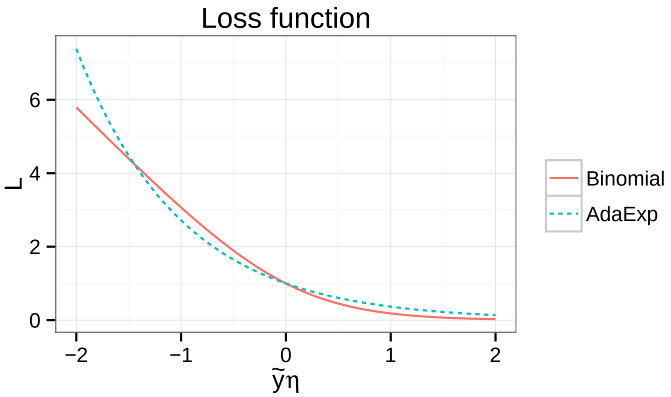


Figure 2.4: The negative binomial log-likelihood loss and the exponential loss as functions of the marginal values $\tilde{y}\eta$. Since $\tilde{y} \in \{-1, 1\}$, a positive product between $\tilde{y}$ and half the estimated log-odds ratio η means correct categorical discrimination.

Alternatively, one can also use the exponential loss function $L(y, \eta) = \exp(-\tilde{y}\eta)$. This basically leads to the famous AdaBoost algorithm by Freund and Schapire (1996). As can be seen in Figure 2.4, this loss function is similar to the negative binomial log-likelihood loss.

2.4.3 Additional Parameters in the Loss Function

The estimation algorithm presented so far is applicable in various circumstances, mainly due to the freedom in the choice of the loss function. We could easily use the same algorithm for count data by postulating the negative Poisson log-likelihood with the natural link function $\log(\mu) = \eta$ as the loss function. Alternatively, the negative binomial distribution can be used to model overdispersed data by utilizing the negative of its density as the loss function. Note, however, that this distribution introduces one additional parameter, which extends the algorithm by a further optimization step. The extra parameter that accounts for overdispersion is optimized additionally within each boosting iteration m. This means that after Equation (2.5 b), one minimizes the empirical risk w.r.t. the overdispersion parameter δ given the current boosting estimate $\hat{\gamma}_{\hat{s}}^{[m]}$

$$\hat{\delta} = \arg\min_{\delta} \sum_{t=1}^{T} L\left(y_t, \hat{\eta}(z_{t,j}; \hat{\gamma}_{\hat{s}}^{[m]}, \delta)\right). \qquad (2.5\text{ c})$$

A thorough treatment of this additional tweak in the algorithm can be found in Schmid, Potapov, Pfahlberg, and Hothorn (2010).

We may even go further, by using gradient boosting for this additional parameter δ. It can be dynamically updated in parallel with the already progressing γ optimization. Therefore, two boosting procedures run in parallel for each parameter and this could be extended to arbitrarily many density parameters $p \in \mathbb{N}$. Given a fixed size of observations T and covariates r, we have an $O(M^p)$ complexity, where M is the optimal step number. This technique was inspired by the Generalized Additive Models for Location Scale and Shape (Rigby and Stasinopoulos, 2005) and proposed by Mayr, Fenske, Hofner, Kneib, and Schmid (2012). Strategies for how to estimate an optimal stop number are proposed in the next section.

2.5 Discussion

This section is intended to briefly discuss the remaining component pieces of boosting. The choice of topics is largely a personal one, with an emphasis on the existing problems of componentwise gradient boosting. Several relevant aspects—deferred in the discussion so far—are addressed: the stopping condition; the regularization amount; the model's parsimony; inference; and others. We also review several improvements of the boosting algorithm and elucidate existing connections to other methods. Some difficulties which prohibit statistical inference, but also references to existing theoretical solutions, are included as well. Finally, we summarize the merits of boosting.

2.5.1 Stopping condition

The number of updates M is the parameter of primary interest in boosting. Each additional boosting step increases the complexity of the model and we should stop reasonably soon in order to avoid overfitting. An intriguing property of boosting is that it overfits slowly. Bühlmann and Yu (2003) showed that the increase in complexity is not linear, and depending on the base learner specification it can diminish exponentially as the iterations grow.

The term *early stopping* refers to the appropriate number of steps after which we solely continue to fit noise. The variable selection property of componentwise boosting crucially depends on early stopping since boosting forever would inevitably include all predictors. Therefore, we generally have two strategies for the final step determination: analytical solution via the model complexity or some numerical device.

The existing analytical solutions are, however, problematic. They use the trace of the hat matrix as a complexity measure, as in the AIC (Akaike, 1973), its corrected version AIC$_c$ (Hurvich, Simonoff, and Tsai, 1998), or the gMDL selection criterion (Hansen and Yu, 2001). The latter is at the heart of Sparse Boosting (Bühlmann and Yu, 2006). The concern is that we do not have an exact model complexity measure as the effective degrees of freedom in the classical regression analysis (Hastie and Tibshirani, 1990; Hastie et al., 2009a).

The hat matrix in boosting is (Bühlmann, 2006; Bühlmann and Hothorn, 2007a)

$$\mathcal{H}_{(m)} = I - (I - \nu\,\mathcal{H}^{[\hat{s}_m]}) \dots (I - \nu\,\mathcal{H}^{[\hat{s}_2]})(I - \nu\,\mathcal{H}^{[\hat{s}_1]}), \tag{2.16}$$

where the $\mathcal{H}^{[\hat{s}_m]}$ are the single hat matrices resulting from the gradient fit at each step m, and I is the identity matrix. Hastie (2007) showed that the complexity measure $\mathrm{df}(m) = \mathrm{tr}(\mathcal{H}_{(m)})$ underestimates the true degrees of freedom due to the selection process. Treating the complexity as if the selected components had been given in advance happens to be highly misleading. The intuition behind this phenomenon is that once several covariates are selected, but still not fully estimated, their estimates are about to randomly progress to the final solutions. The instant model is, therefore, more complex than the formula suggests, and underestimating the true model complexity leads to overfitting (Hastie, 2007). Bühlmann and Hothorn (2007b) propose a correction for the degrees of freedom for linear base learners which takes the number of selected covariates into account. The active set of covariates $\mathrm{df}_{\mathrm{actset}}(m)$ represents the number of covariates selected up to step m. This appears to be a better approximation for the true degrees of freedom, but still, both $\mathrm{df}_{\mathrm{actset}}(m)$ and $\mathrm{df}(m)$ remain random variables and, therefore, cannot be regarded as classical degrees of freedom.

The determination of an optimal step number M can be done via some cross-validating assessment and this is the recommended stop solution. First, we choose a sufficiently large M_{large} that is expected to fulfill $M < M_{\mathrm{large}}$. Then, we resample the data set using cross-validation, bootstrapping, or subsampling, to name a few. With k-fold cross-validation we split the data set into k disjoint, equally large parts. We leave one part for validation (validation sample) and fit the boosting model with M_{large} steps on the remaining data points (learning sample). We do this k times, until each part has been used as validation sample. As a result, we collect k out-of-sample loss function estimates for the steps 1 through M_{large}. The optimal step M is the one with the smallest empirical loss on average. The idea behind subsampling is similar to cross-validation, with the sole difference that the parts are allowed to intersect and in bootstrapping we sample with replacement.

2.5.2 Regularization

Imputing a penalty to the parameters is a common regularization device for variable selection. The benefit of shrunken parameters is usually observable in the improved out-of-sample predictive accuracy. For example, penalizing the number of parameters is used in many information criteria, such as the AIC or BIC. Another example is Lasso (Tibshirani, 1996), which minimizes the L_2 loss function with an L_1 penalty, but it is generally unclear which regularization criterion optimizes boosting (Zhao and Yu, 2007). A small step size ν coupled with early stopping is the regularization device of boosting. This algorithmic constraint leaves many mathematical questions open.

For very special cases, however, there exist some theoretical explanations. Efron, Hastie, Johnstone, and Tibshirani (2004) showed that having orthogonal predictors and linear base learners, boosting with infinitesimally small step-size, or the forward stagewise linear regression[4] (FSLR) as they call it, is equivalent to the Lasso. Bühlmann and van De Geer (2011, p. 388) state that:

> There is a striking similarity between gradient based boosting and the Lasso in linear or generalized linear models. Thus, despite substantial conceptual differences, boosting-type algorithms are implicitly related to L_1-regularization.

This is also confirmed by Zhao and Yu (2007), who show that in even more general situations, one can get the Lasso solution via boosting. They implement an additional backward elimination step which removes the irrelevant variables accumulated by the forward run.

Meinshausen and Bühlmann (2010) address the problem of proper regularization in variable selection methods with their method, called *stability selection*. Stability selection is not an alternative model for variable selection but it is rather an additional step which extends already existing selection methods. The main idea is to randomize the learning algorithm by subsampling the original data and simulating the selection probabilities of all variables. From these, only the most frequent variables—above some threshold level—are included in the final model. The authors show that the method is not sensitive to a reasonably varied threshold level. They also provide an upper bound for the false discovery rate, which means that one can control the number of false selections. The upper bound, however, is guaranteed under the nontrivial assumption that the false variables compete among themselves at

[4] FSLR is a slightly modified, scale-variant version of boosting for linear models.

random, i.e., their distribution is exchangeable (Meinshausen and Bühlmann, 2010, Theorem 1 and the discussion).

2.5.3 Bias

Stronger regularization leads to sparser and better interpretable models. This property, however, is in conflict with the objective of an unbiased parameter estimation. The latter is sacrificed by boosting for the efficiency of model interpretability and out-of-sample prediction.[5]

The parameter estimates made by boosting are typically underestimated due to the parameter regularization via early stopping. The algorithm starts with initial zero estimates, gradually increases them, and stops before convergence. This is typical for shrinkage methods in finite samples, where the parameters usually have smaller magnitudes than the unregularized solutions and the bias vanishes with an increasing sample size. Such methods prove to show better out-of-sample performance than fully estimated, unregularized models.

Biased estimates are, therefore, typical for high-dimensional models in which variable selection is desired. If not stopped early, the algorithm converges to the full parameter estimates. Achieving both variable selection and unbiased parameter estimations can be heuristically set up in a two-step procedure: first regularize in order to detect influential variables and second allow for fully converged estimates with these selections. In the same vein Fan and Lv (2008) propose independence screening to reduce the computation in ultra-high dimensional variable selection (see also Fan and Lv, 2010, for further details). One should, however, acknowledge that the estimation of finite-sample distributions of such *post-model-selection estimators* is infeasible (Leeb and Pötscher, 2005).

2.5.4 Forwardness

Boosting only works in a forward fashion. It falls into the category of the so called *greedy* methods which strive to reduce the empirical loss at each step and are not able to adjust the previous steps. Forward selection methods, which are also greedy, tend to be unstable in out-of-sample prediction (Breiman, 1996).

[5] One algorithm that satisfies both properties is the smoothly clipped absolute deviation (SCAD) model proposed by Fan and Li (2001).

This is especially true when the predictors are correlated. As an explanatory example, Efron et al. (2004) take the forward stepwise regression (Weisberg, 1980), but the intuition is similar for more general methods, e.g., the backfitting algorithm (Hastie and Tibshirani, 1990). Forward stepwise regression performs simple linear regression of the response on the predictor with the highest correlation in magnitude. After the fit, the new residual vector substitutes the response and the algorithm seeks for the next correlated candidate. The previously selected predictor is therefore excluded, since it is now orthogonal to the new response, but so are all predictors which are correlated with it. Therefore, the algorithm automatically eliminates them from the selection process.

The main difference between forward stepwise regression and forward stagewise regression (or boosting) is that the greediness of the first algorithm is somewhat reduced by the inclusion of the shrinkage factor in the second. Boosting gradually blends the predictors in the model instead of including them all at once. Therefore, even correlated predictors can compete for inclusion in the model as long as they are equally correlated with the gradient.

2.5.5 Sparsity

Models are sparse if only a small subset of the potential variables are truly relevant. Sorting out the substantial variables is beneficial both for interpretation and for prediction, and is the main reason why sparse models are so appealing. Boosting solutions are sparse, but in linear settings not as sparse as Lasso for example (Zhao and Yu, 2007). Therefore, some additional modifications are necessary in order to achieve sparser models.

A modification of boosting with better selection properties is the *twin boosting* proposed by Bühlmann and Hothorn (2010). Twin boosting consists of two rounds. The first one is the classical componentwise boosting as discussed so far. The second round takes the parameter estimates from the first round into account and rescales the components in the selection process (2.5 b). Roughly speaking, the correlation between the gradient and the components is increased proportionally to the estimates, so that components with bigger coefficients are preferred for selection. They show that for special cases the selection of twin boosting is equivalent to the adaptive Lasso proposed by Zou (2006).

The aforementioned stability selection method (Meinshausen and Bühlmann, 2010) generally leads to sparser models in Lasso but is not always better for boosting than trivial cross-validation. Further, the backward step in stagewise Lasso (also called BLasso from boosting Lasso, Zhao and Yu, 2007) leads to sparser models when compared to the classical linear boosting.

2.5.6 Inference

The mathematical theory for high-dimensional statistics is still largely under development. An excellent overview of the computational and mathematical advances in this field can be found in Hastie et al. (2009a); Fan and Lv (2010) Bühlmann and van De Geer (2011). Boosting, in particular, is a provably consistent estimator for linear regression models (Zhang and Yu, 2005; Bühlmann, 2006).

Classical statistical inference, however, cannot be applied to high-dimensional problems. One technical reason is that having more variables than observations[6] leads to a lower-rank design matrix and consequently to ill-posed test problems. Proper statistical inference is also hindered by the estimation bias and the lack of appropriate degrees of freedom as previously discussed.

The culprit is the selection process, whose result cannot be regarded as if it had been given in advance. Earlier, we mentioned the negative impact of this assumption on the model's complexity, but it can be even worse with respect to the finite sample distributions of the estimators. Leeb and Pötscher (2005) show that even relying on a consistent model selection procedure, we cannot simply use the standard asymptotic distributions which we would have applied in the absence of model selection. Furthermore, the authors warn that the true asymptotic properties of these estimators heavily depend on the true, unknown parameter values. Therefore, it is by no means guaranteed the asymptotics occur at all, regardless of the sample size. On the other hand, the finite-sample distributions of the estimators are typically complicated, but their dependence on the unknown parameters motivates Leeb and Pötscher (2005, p. 23) to make the following statement:

> Estimation of these finite-sample distributions is "impossible" (even in large samples). No resampling scheme whatsoever can help to alleviate this situation.

As a final remark, it should be mentioned that under the assumptions of sufficiently large parameter values and orthogonal design, the concerns raised by Leeb

[6] In componentwise boosting one can get a larger number of selected covariates than observations. In contrast, Lasso allows at most $\min(T, r)$ non-zero parameters (Bühlmann and van De Geer, 2011).

and Pötscher (2005) no longer apply. Moreover, the bootstrap provides a consistent estimator for the finite-sample distributions of the estimators and is commonly used to quantify the variability of the estimators in practice.

2.5.7 Base learners

The generic structure of gradient boosting allows any regression-type statistical model to be used as a base learner, e.g., linear regression, logistic regression, (bivariate) P-spline regression, classification, regression trees, and many more. A detailed overview and practical implementation (with R) of several base learners can be found in Hofner (2012) and Hofner et al. (2012). Having particular problems at hand, we will present several base learners in the subsequent sections.

As a final note, we summarize the merits of componentwise gradient boosting.

Component selection: Least open to objection is the need for a statistical technique to select the most informative variables out of a large set of predictors. Complex financial markets, modeled by an exhaustive set of macro and financial drivers, offer a challenge in sorting out irrelevant predictors, or some of their lags. For example, the price or the volatility of a stock depends not only on its past values, but also on the past values of other, exogenous variables. Therefore, the number of variables that influence asset prices or their volatility can be huge. Furthermore, selecting the relevant predictors when their number exceeds the number of observations is surely a nontrivial task.

Model selection: A related problem is to select the functional dependence between the relevant variables and the response (Kneib et al., 2009). This is called model choice and occurs in most regression-type problems, regardless of the number of variables. For example, a continuous covariate could be included in a statistical model using linear, non-linear, or interaction effects with other predictors.

Versatility: A generalized additive model is specified as the combination of a distributional assumption and a structural assumption. The distributional assumption specifies the conditional distribution of the outcome through the loss function (see Section 2.4). The structural assumption specifies the types of effects that are to be used in the model (base learners). The loss function is independent of the estimation of the base learners, hence one can freely combine structural and distributional assumptions to tackle new estimation problems. This modular specification of the dependence between any distributional characteristic and some, initially unknown, subset of predictors is what makes boosting versatile and a realistic problem solving utility.

Interpretability: Even though in statistics it is natural and obvious for a prediction model to be interpretable, in machine learning this is less true. "Black box" algorithms such as bagging, and especially random forests (Breiman, 2001a), are prominent examples of strong prediction algorithms which lack interpretatibility. Due to the additive update in (2.9), the estimate of a function f_j at the final iteration M has the same structure as the corresponding base learner. Depending on the circumstances, we could easily combine established statistical models into one, and the structural assumption of the model will be specified, and therefore perfectly interpretable, by the base learners.

Prediction accuracy: Boosting is optimized with respect to the out-of-sample predictive accuracy. This, in addition to the shrinking effect of the estimates towards zero, usually results in a strong forecasting performance.

Chapter 3

Boosting Techniques for Nonlinear Time Series Models

Many of the popular nonlinear time series models require *a priori* the choice of parametric functions which are assumed to be appropriate in specific applications. This approach is mainly used in financial applications, when sufficient knowledge is available about the nonlinear structure between the covariates and the response. One principal strategy to investigate a broader class on nonlinear time series is the Nonlinear Additive AutoRegressive (NAAR) model. The NAAR model estimates the lags of a time series as flexible functions in order to detect non-monotone relationships between current and past observations. Robinzonov, Tutz, and Hothorn (2012) consider linear and additive models for identifying nonlinear relationships which is presented in this chapter. A componentwise boosting algorithm is applied for simultaneous model fitting, variable selection, and model choice. Thus, with the application of boosting for fitting potentially nonlinear models we address the major issues in time series modeling: lag selection and nonlinearity. By means of simulation we compare boosting to alternative nonparametric methods. Boosting shows a strong overall performance in terms of precise estimations of highly nonlinear lag functions. The forecasting potential of boosting is examined on German industrial production (IP); to improve the model's forecasting quality we include additional exogenous variables. Thus we address the second major aspect in this chapter which concerns the issue of high-dimensionality in models. Allowing additional inputs in the model extends the NAAR model to a broader class of models, namely the NAARX model. We show that boosting can cope with large models which have many covariates compared to the number of observations.

3.1 Introduction

In modeling of times series we often deal with two issues, linear modeling in case of nonlinear structures, and high-dimensionality. Boosting is a way to address both. Linear time series models encounter various limitations and are applicable only under very restrictive conditions. Some of these constraints have been relieved in the past two decades. In particular, the nonparametric regression was adapted for time series, allowing more flexibility than linear modeling (e.g., Lewis and Stevens, 1991; Chen and Tsay, 1993; Huang and Yang, 2004). A leading aspect to be explored throughout this chapter is the nonparametric modeling of time series and the resulting forecasting techniques.

The second major aspect concerns the issue of high-dimensionality in the models, i.e., models taking potentially many covariates into account. Boosting, one of the most influential strategies that deal with high-dimensional models, has its roots in machine learning. The idea has undergone significant evolution in the past decade. It has been successfully applied to statistical model fitting (e.g., Bühlmann and Hothorn, 2007a). Audrino and Bühlmann (2003) are the first to introduce boosting in a financial context. They apply boosting with tree based learners for volatility estimation of heteroskedastic time series. Audrino and Bühlmann (2009) further propose boosting with multivariate B-splines for volatility estimation in a heteroskedasticity type of model. Boosting of GARCH models can be found in Audrino and Barone-Adesi (2006), Matías, Febrero-Bande, González-Manteiga, and Reboredo (2010) among others. In the present work we focus on lag selection, detection of nonlinear relationships between the return, its lagged values and exogenous components, as well as forecasting.

Due to the frequent use of the simple univariate autoregressive model (AR), we draw on it as a benchmark in the application part to follow. For a substantially broader discussion on times series, see Hamilton (1994). In addition we consider the vector autoregressive (VAR) model. The VAR model suggests that every variable is a linear combination of its past observations and the past observations of supplemental variables. In practice such assumptions enjoy great popularity. Multivariate time series are considered in greater depth by Lütkepohl (1991; 2006).

The literature offers a great amount of nonlinear modeling tools. Many of them are developed in the spirit of nonlinear parametric models. They require an a priori choice of parametric functions, which are assumed to be appropriate in specific situations. That approach is used mainly in financial applications, when sufficient knowledge is available about the nonlinear structure between the covariates and the response. However, the appropriateness of such assumptions is usually hard to justify in practice.

In contrast to parametric nonlinear models, nonparametric techniques are not restricted to a particular choice of parametric functions. One principal strategy is to study the times series counterpart of the additive model; the so-called Nonlinear Additive AutoRegressive (NAAR) model (Chen and Tsay, 1993). When further (exogenous) variables are available, we suitably extend the model with more functions and call it NAARX (Chen and Tsay, 1993). Thus, NAARX encompasses linear regressive models and many nonlinear models as special cases.

The literature on nonlinear additive models is extensive, therefore, we concentrate on nonparametric approaches. Huang and Yang (2004) introduced a method that attracted much attention because of appealing lag-selection properties for univariate nonlinear time series. It essentially represents an additive version of the linear stepwise procedure using truncated splines, or B-splines, as base expansions of the predictors. The proposed base functions are not penalized. Instead, a formula is suggested which determines a relatively small number of evenly spaced knots. In terms of lag selection, the proposed method performed quite well with simulated time series. However, no results were provided that show the goodness-of-fit of the models. We will use some of the artificial times series, provided by Huang and Yang (2004) in Section 3.3 and will shed light upon the goodness-of-fit as well.

Multivariate Adaptive Regression splines (MARS) were introduced by Friedman (1991). An excellent overview of the method is available in Hastie et al. (2009a, Chapter 9), an application of MARS in a time series context is provided by Lewis and Stevens (1991). The last nonparametric model that we consider is the BRUTO procedure (Hastie and Tibshirani, 1990, Chapter 9). BRUTO combines inputs selection with backfitting by using smoothing splines. It was applied to time series by Chen and Tsay (1993). See Hastie and Tibshirani (1990, p. 90–91) for details concerning backfitting and Hastie and Tibshirani (1990, p. 262) for the BRUTO algorithm.

We proceed as follows. In Section 3.2, we introduce the general ideas behind our model. Exemplified by two different types of base learners, we examine the structure of the boosting algorithm for continuous data. The first base learner is the simple linear model, the second one is a penalized B-spline (Eilers and Marx, 1996). Section 3.3 examines the results of a simulation study. We analyze the performance of boosting with P-spline base learners in Monte Carlo simulations with six artificial, nonlinear, autoregressive time series. We compare the outcomes of boosting to the outcomes obtained through alternative nonparametric methods. Their performances are considered in terms of lag selection and goodness-of-fit. In Section 3.4 we apply boosting with both learners to real world data in terms of a direct forecasting. The target variable is German industrial production. We compare boosting, along with other methods, to the simple univariate autoregressive model.

3.2 The Model

The statistical framework developed by Friedman (2001) interprets boosting as a method for direct function estimation. He shows that boosting can be interpreted as a basis expansion, in which every single basis term is iteratively refitted. Still, some care must be taken in interpreting boosting as a basis expansion. In contrast to conventional basis expansions, where the basis functions are known in advance, the basis's members and also their number are iteratively determined by the fitting procedure. Our notation is as follows:

$$\mathbf{z}_t = (\mathbf{y}_t^\top, \mathbf{x}_t^\top)^\top = (y_{t-1}, \dots, y_{t-p}, x_{t-1}^{(1)}, \dots, x_{t-p}^{(1)}, \dots, x_{t-1}^{(q)}, \dots, x_{t-p}^{(q)})^\top \in \mathbb{R}^{(q+1)p}$$

denotes the p-lagged vector of explanatory variables representing the lagged values $\mathbf{y}_t = (y_{t-1}, \dots, y_{t-p})^\top \in \mathbb{R}^p$ of the endogenous variable $y_t \in \mathbb{R}$ and the lagged values of q exogenous variables $\mathbf{x}_t \in \mathbb{R}^{qp}$. The proposed model is then

$$\begin{aligned}\mathbb{E}(y_t|\mathbf{z}_t) &= \sum_{i=1}^{p} f_i(y_{t-i}) + \sum_{i=1}^{p} f_i^{(1)}(x_{t-i}^{(1)}) + \dots + \sum_{i=1}^{p} f_i^{(q)}(x_{t-i}^{(q)}) \\ &= \sum_{i=1}^{p} f_i(y_{t-i}) + \sum_{j=1}^{q}\sum_{i=1}^{p} f_i^{(j)}(x_{t-i}^{(j)}) =: \eta(\mathbf{z}_t). \qquad (3.1)\end{aligned}$$

The objective is to obtain an estimate $\hat{\eta}$ of the function η. With real data one wants to minimize

$$\hat{\eta} = \arg\min_{\eta} \frac{1}{T} \sum_{t=1}^{T} L(y_t, \eta(\mathbf{z}_t)). \qquad (3.2)$$

where L is some loss function. One of the frequently employed loss functions is the squared-error loss L_2

$$L(y_t, \eta(\mathbf{z}_t)) = \frac{1}{2}(y_t - \eta(\mathbf{z}_t))^2, \qquad (3.3)$$

which is also chosen in this work. A discussion of the specification of several loss functions can be found in Section 2.4, as well as in Hastie et al. (2009a, chap. 10), Bühlmann and Hothorn (2007a), Friedman (2001), and in particular in Lutz, Kalisch, and Bühlmann (2008). We further introduce the parameters $\boldsymbol{\beta}$ that will facilitate interpretation later on and reformulate the problem as

$$\hat{\eta} = \eta(\,\cdot\,; \hat{\boldsymbol{\beta}}) = \arg\min_{\eta} \frac{1}{T} \sum_{t=1}^{T} L(y_t, \eta(\mathbf{z}_t; \boldsymbol{\beta})). \tag{3.4}$$

The final solution of (3.4) is expressed in terms of a sum over M base learners h, the mth of which depends on a parameter vector $\hat{\boldsymbol{\gamma}}^{[m]}$:

$$\eta(\,\cdot\,; \hat{\boldsymbol{\beta}}^{[M]}) = \sum_{m=0}^{M} \nu h(\,\cdot\,; \hat{\boldsymbol{\gamma}}^{[m]}) \tag{3.5}$$

where $\hat{\boldsymbol{\gamma}}^{[0]}$ is an arbitrary chosen start vector of parameters, $\nu \in (0,1)$ is the shrinkage parameter and the parametric function h represents the base learner. See Chapter 2 for details regarding the optimization of (3.4). Further, we specify the base learners $h(\,\cdot\,; \hat{\boldsymbol{\gamma}}^{[m]})$.

3.2.1 Componenwise Linear Base Learner

When many predictors are available, a fruitful strategy is componentwise boosting. Originally proposed by Bühlmann and Yu (2003) and further developed by Bühlmann (2006), the key idea of this method is to exercise the base learner upon *one* variable at a time and to pick out only this component with the largest contribution to the fit. Thus, we keep the learner “weak” enough by restraining a complex structure with many parameters.

The simplest base learner is linear. For this learner $\hat{\boldsymbol{\gamma}}^{[m]} = (0, \ldots, \hat{\gamma}_{\hat{s}_m}, \ldots, 0)^\top$ is a $(q+1)p$-dimensional vector with zeros for all but the $\hat{s}_m$th component, where $\hat{s}_m \in \{1, 2, \ldots, (q+1)p\}$ denotes the respective component at the mth boosting step. The definition of the base learner is as follows:

componentwise linear base learner

$$h(\mathbf{z}_t; \hat{\boldsymbol{\gamma}}^{[m]}) = \mathbf{z}_t^\top \hat{\boldsymbol{\gamma}}^{[m]}, \text{ where } \hat{\boldsymbol{\gamma}}^{[m]} = (0, \ldots, \hat{\gamma}_{\hat{s}_m}, \ldots, 0)^\top \in \mathbb{R}^{(q+1)p},\ \hat{\gamma}_{\hat{s}_m} \in \mathbb{R}$$

$$\hat{\gamma}_j = \text{OLS}(\gamma_j), \qquad \forall j \in J := \{1, 2, \ldots, (q+1)p\} \tag{3.6}$$

$$\hat{s}_m = \arg\min_{j \in J} \sum_{t=1}^{T} (g^{[m]}(\mathbf{z}_t) - h(\mathbf{z}_t; \hat{\boldsymbol{\gamma}}^{[j]}))^2, \tag{3.7}$$

where $\text{OLS}(\gamma_j)$ is the Ordinary Least Squares Estimator of γ_j with the negative gradient being used as a pseudo-response. Thus, the base procedure fits a simple linear regression $(q+1)p$ times as shown in (3.7), and the chosen component $\hat{s}_m$ is the one which fits to this pseudo-response best. We refer to this procedure as GLMBoost later on.

3.2.2 Componentwise P-spline Base Learner

We now refer to the flexible structure defined in (3.1) and employ P-splines with evenly spaced knots as base learners. That means that the base learner is represented by a Generalized Additive Model with P-splines (Eilers and Marx, 1996). Note that the term additive expansion can be used in two different contexts. Here we suggest an initial additive expansion of the covariates, which should be clearly distinguished from the interpretation of boosting as an additive expansion itself. Thus, the f's in (3.1) are represented by the sum of B known basis functions b_l, $l = 1, \dots, B$.

In the previous section we defined a componentwise selection of linear predictors. In the current section, likewise, we do the same with more flexible learners. The essential modifications concern $\hat{\boldsymbol{\gamma}}^{[m]} = (\mathbf{0}^\top, \dots, \hat{\boldsymbol{\gamma}}_{\hat{s}_m}^\top, \dots, \mathbf{0}^\top)^\top \in \mathbb{R}^{(q+1)pB}$ having qpB zeros, $\hat{\boldsymbol{\gamma}}_{\hat{s}_m} = (\gamma_1, \dots, \gamma_B)^\top \in \mathbb{R}^B$ and $\mathbf{0} = (0, \dots, 0)^\top \in \mathbb{R}^B$ and the base learner being a P-spline instead of a straight line. Subsequently, the estimations $\hat{\boldsymbol{\gamma}}_{\hat{s}_m}$ are obtained through the penalized least squares estimator and not through the OLS-Estimator. The base procedure is as follows:

componentwise P-spline base learner

$$h(\mathbf{z}_t; \hat{\boldsymbol{\gamma}}^{[m]}) = \mathbf{Z}_t^\top \hat{\boldsymbol{\gamma}}^{[m]}$$

$$\hat{\boldsymbol{\gamma}}_j = \text{PLSE}(\boldsymbol{\gamma}_j), \qquad \forall j \in J := \{1, 2, \dots, (q+1)p\} \tag{3.8}$$

$$\hat{s}_m = \arg\min_{j \in J} \sum_{t=1}^{T} (g^{[m]}(\mathbf{z}_t) - h(\mathbf{z}_t; \hat{\boldsymbol{\gamma}}^{[j]}))^2 \tag{3.9}$$

where $\mathbf{Z}_t \in \mathbb{R}^{(q+1)pB}$ is the basis expansion of $\mathbf{z}_t$, $\text{PLSE}(\boldsymbol{\gamma}_j)$ is the Penalized Least Squares Estimator of $\boldsymbol{\gamma}_j$ with the negative gradient being used as a pseudo response. This procedure is referred to as GAMBoost.

Essentially, we estimate two components at each stage: all candidate parameters for the update (3.8), and the index of the "best" candidate (3.9). Since the negative gradient indicates the direction of the locally greatest decrease in loss, the most "valuable" covariate has the highest correlation with the negative gradient and is therefore chosen for fitting. The final model fit typically depends on a subset of the original $(q+1)p$ covariates.

The price for increased flexibility is the inclusion of additional parameters. One should consider not only an appropriate stopping value M and a shrinkage factor ν, but also a smoothing parameter λ, and a number of evenly spaced knots. Schmid and Hothorn (2008) carried out an analysis of the effect of these parameters and showed that M is essentially the single parameter that matters. All others are regarded as hyper-parameters since the algorithm is robust to their alteration. It is worth emphasizing the effect of λ for determining the degrees of freedom (df) of the weak learner. High values of λ lead to low degrees of freedom which is preferable in order to keep the learner highly biased but with a low variance. Schmid and Hothorn (2008) proposed df $\in [3,4]$ as a suitable amount for the degrees of freedom. We follow these prescriptions and remind that the reasonable altering of this parameter reflects solely in the computational time.

3.3 Simulation Study

In this section we investigate the performance of boosting an additive model in Monte Carlo simulations with six artificial nonlinear autoregressive time series. We compare the outcomes of boosting to the outcomes obtained through alternative nonparametric methods. These are the method by Huang and Yang (2004), referred to as the acronym HaY, BRUTO, and MARS. Their performance is considered in terms of in-sample goodness-of-fit. The dynamics of the simulated processes are shown in Table 3.1 where ϵ_t are independent and identically distributed $N(0,1)$ random variables. NLAR1U1 and NLAR1U2 have one lag and were used by Huang and Yang (2004). Besides, there are three models with two lags: NLAR2b-NLAR2d. All but NLAR2c two-lag models were originally used by Tschernig and Yang (2000), NLAR2c was used by Chen and Tsay (1993). The last NLAR4 model has four lags and was used by Shafik and Tutz (2009).

All models from Table 3.1 have been simulated 100 times with sizes $400+N$, the first 400 values discarded and $N = p+T$, with $p = 10$ pre-sample values and $T = 50, 100, 200$ in-sample observations. Such partitioning of the time series values is convenient in order to ensure same sample size of T for each covariate at a given period and to simplify the notation. As p suggests, the maximum lag length has been limited to ten.

Table 3.1: Dynamics of six artificial time series.

Model	Function
NLAR1U1	$y_t = -0.4(3 - y_{t-1}^2)/(1 + y_{t-1}^2) + 0.1\epsilon_t$
NLAR1U2	$y_t = 0.6(3 - (y_{t-2} - 0.5)^3)/(1 + (y_{t-2} - 0.5)^4) + 0.1\epsilon_t$
NLAR2b	$y_t = (0.4 - 2\exp(-50y_{t-6}^2))y_{t-6} + (0.5 - 0.5\exp(-50y_{t-10}^2))y_{t-10} + 0.1\epsilon_t$
NLAR2c	$y_t = 0.8\log(1 + 3y_{t-1}^2) - 0.6\log(1 + 3y_{t-3}^2) + 0.1\epsilon_t$
NLAR2d	$y_t = (0.4 - 2\cos(40y_{t-6})\exp(-30y_{t-6}^2))y_{t-6} \quad +$ $(0.55 - 0.55\sin(40y_{t-10})\sin(40y_{t-10}))\exp(-10y_{t-10}^2) + 0.1\epsilon_t$
NLAR4	$y_t = 0.9((\pi/8)y_{t-4}) - 0.75\sin((\pi/8)y_{t-5}) + 0.52\sin((\pi/8)y_{t-6}) +$ $0.38\sin((\pi/8)y_{t-7}) + 0.1\epsilon_t$

In simulations we can measure how precisely a fitting procedure reflects the true dynamics of a simulated process. In case of a linear time series, a convenient measure is the Euclidian distance between the true parameter vector and the estimated one. When dealing with nonparametric models, we need a more complex accuracy measure for the discrepancy between functions. We consider the squared residuals between the true partial functions (or lag functions) centered to mean zero and the estimated functions.

Let $\tilde{f}_k$ denote the kth true lag function after centering it to mean zero, i.e., subtracting its mean value. Then the mean squared prediction error is

$$\text{MSPE}_k = \frac{1}{n}\sum_{i=1}^{n}[\tilde{f}_k(z_i) - \hat{\tilde{f}}_k(z_i)]^2 \tag{3.10}$$

where $\hat{\tilde{f}}_k$ is the estimated counterpart of $\tilde{f}_k$. We choose a total number of $n = 200$ evenly spaced observations z_i which are located between the 5th and 95th quantile of the empirical distribution of y_{t-k}. The accuracy measure is the average of the individual MSPE's

$$\text{MSPE} = \frac{1}{p}\sum_{k=1}^{p}\text{MSPE}_k. \tag{3.11}$$

Figures 3.1 and 3.2 reflect a typical result of the estimation repetitions. We depict the true process dynamics (circled lines) along with the estimated ones (solid lines). This visual excerpt gives the satisfying impression of boosting being capable of discovering the truth.

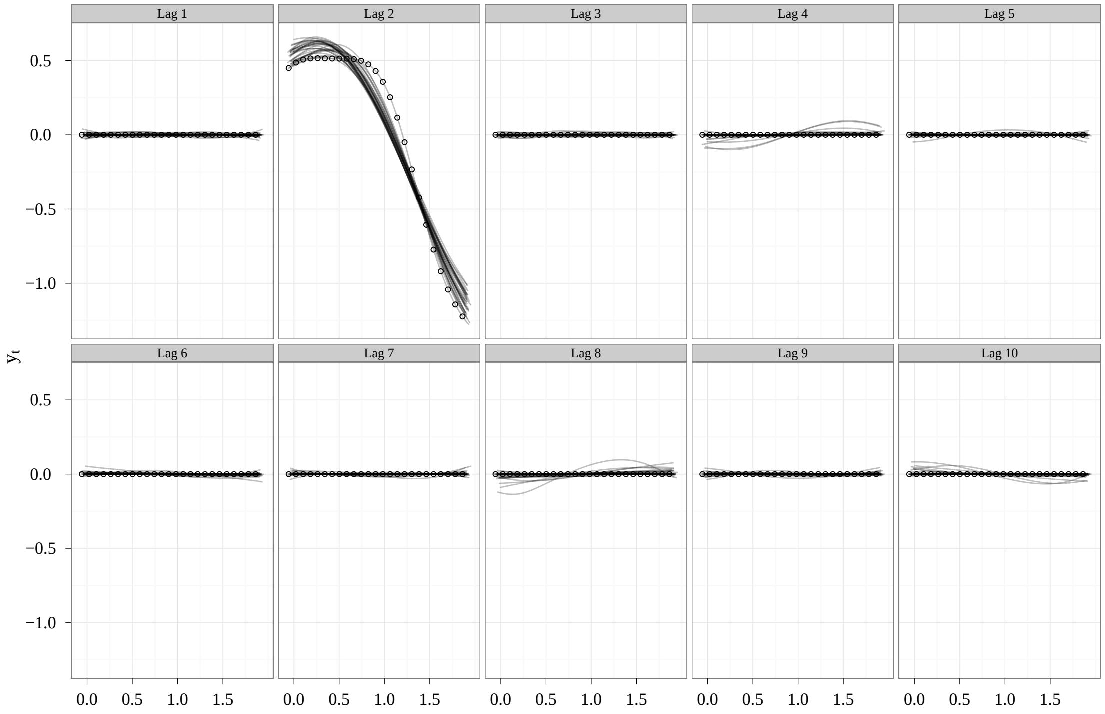


Figure 3.1: Boosting estimations of the lag functions of NLAR1U2. True lag is 2 (circled line), estimated lags are depicted as solid lines. The functions are mean zero centered.

The results of the median MSPE across all 100 simulation runs are summarized in Table 3.2. The rows contain the simulated series, the columns represent the different modeling techniques. NLAR1U and NLAR1U2 yield the most parsimonious models. Their dynamics seems to be explained very well by MARS, HaY and GAMBoost, while BRUTO performed very poorly. For NLAR1U2, we notice that despite overfitting in sense of selected lags, boosting estimated the relevant function quite precisely, e.g., $T = 50, 100$. This suggests that the redundant functions were considered close to zero. It is reassuring to see the apparently zero estimations of the redundant lags in NLAR1U2 (Figure 3.1).

The literature on nonparametric regression for dependent data is relatively sparse, especially when related to boosting. Strong serial dependence might mislead the fitting procedure to produce erroneous transformations. For instance, this is evident for boosting of NLAR2c, shown in Figure 3.2, where the second and the seventh lag

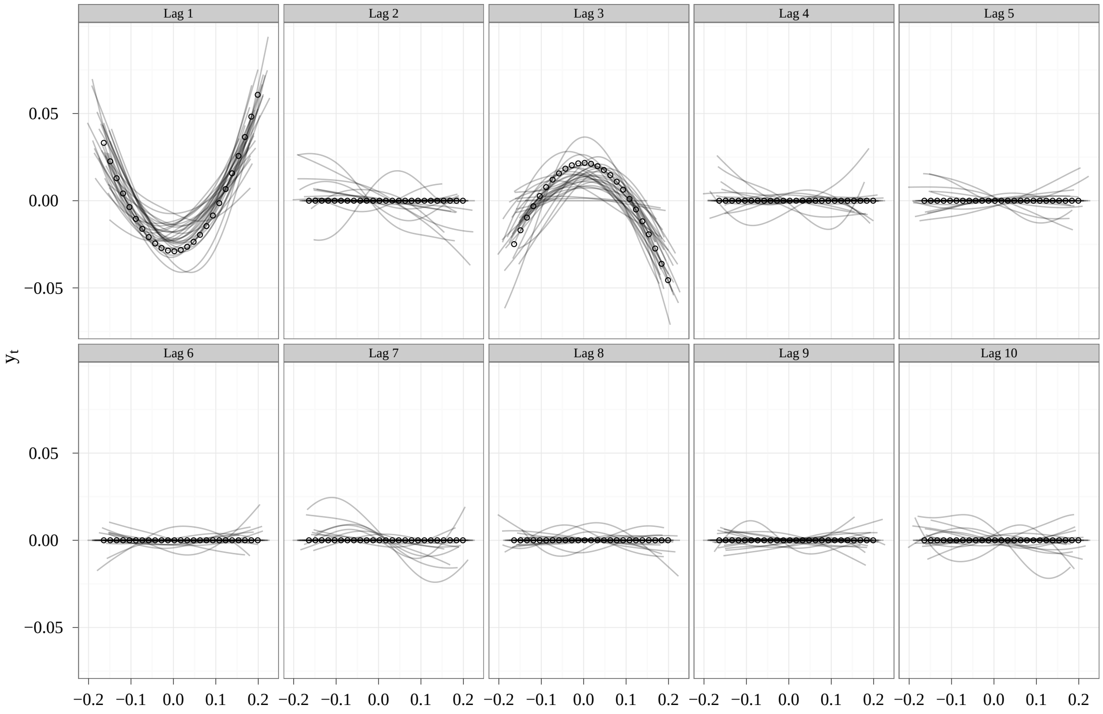


Figure 3.2: Boosting estimations of the lag functions of NLAR2c. True lags are 1 and 3 (circled lines), estimated lags are depicted as solid lines. The functions are mean zero centered.

were overfitted rather strongly.

With an increasing number of significant covariates both BRUTO and GAMBoost improved their performance. The boxplots shown in Figure 3.3 propose a visual confirmation of this observation. They represent the MSPE of each modeling strategy which occurred throughout the repetitions. The exclusion of significant covariates by the non-boosting methods was, on balance, more counterproductive than the inclusion of redundant ones by boosting. GAMBoost showed, overall, strong estimation properties. It was superior to its rivals in the larger model specifications and was evidently competitive even in the small ones. It is worth mentioning, that in the small sample sizes the advantage of boosting was more evident. Therefore, GAMBoost showed good prediction accuracy when the information content of the data decreased, i.e., where a low signal-to-noise ratio was observed.

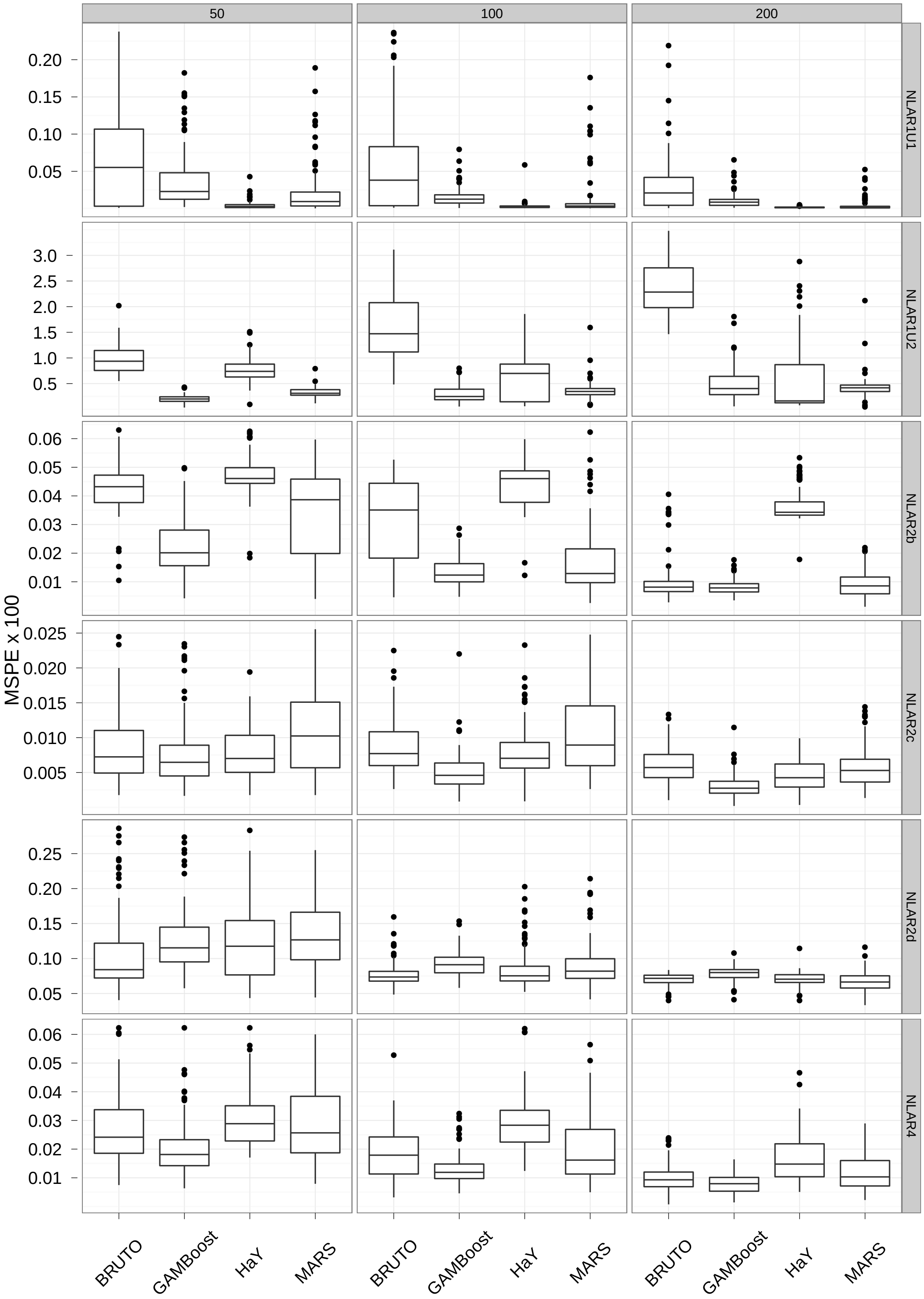


Figure 3.3: Boxplots of the Monte-Carlo simulations.

Table 3.2: Simulation results of the median MSPE of 100 simulation runs multiplied by 100. Boldface numbers indicate the best model performance for each setup.

Model	T	GAMBoost	BRUTO	MARS	HaY
NLAR1U1	50	0.0228	0.0895	0.0093	**0.0027**
	100	0.0141	0.0508	0.0039	**0.0020**
	200	0.0080	0.0278	0.0016	**0.0014**
NLAR1U2	50	**0.4035**	2.5098	0.4288	0.7184
	100	**0.2380**	1.6916	0.3381	0.7289
	200	0.1789	0.9420	0.3049	**0.1622**
NLAR2b	50	**0.0201**	0.0443	0.0393	0.0470
	100	**0.0123**	0.0349	0.0140	0.0455
	200	**0.0074**	0.0084	0.0078	0.0358
NLAR2c	50	**0.0065**	0.0077	0.0120	0.0072
	100	**0.0049**	0.0074	0.0084	0.0067
	200	**0.0028**	0.0054	0.0058	0.0042
NLAR2d	50	0.1154	**0.0886**	0.1375	0.1260
	100	0.0925	**0.0786**	0.0877	0.0766
	200	0.0788	0.0704	**0.0672**	0.0699
NLAR4	50	**0.0181**	0.0247	0.0278	0.0301
	100	**0.0133**	0.0176	0.0197	0.0278
	200	**0.0077**	0.0085	0.0104	0.0147

3.4 Economic Forecasting with Boosting

In this section boosting, along with other parametric and nonparametric models, are applied to real data. The target variable is German industrial production (IP) with 176 observations for the time period 1992:01 – 2006:08. In order to circumvent any structural breaks due to the reunification, the data before 1991 was omitted. Data from 1991 is not included either, because some of the exogenous variables used here, such as ZEW Economic Sentiment, FAZ Indicator, have only been available after 1992. The series was obtained from Deutsche Bundesbank[1] and is seasonally and workday adjusted. The data, as well as the leading indicators from Section 3.4.3, were also used by Robinzonov and Wohlrabe (2010). The exact monthly growth rates are taken to eliminate non-stationarity which is

$$\Delta(\mathrm{IP}_t) = \frac{\mathrm{IP}_t - \mathrm{IP}_{t-1}}{\mathrm{IP}_{t-1}}.$$

Forecasting of IP is frequently performed in practice. Contributions to the forecasting of German industrial production include Hüfner and Schröder (2002), Benner and Meier (2004), Dreger and Schumacher (2005) among others.

Historically, the focus in forecasting has been on low-dimensional univariate or multivariate models, all sharing the common linearity in the parameters. Recently, additional studies exist that investigate the forecasting performance of nonlinear time series models, e.g., Clements, Franses, and Swanson (2004), Teräsvirta, van Dijk, and Medeiros (2005), Claveria, Pons, and Ramos (2007), Elliot and Timmermann (2008). Audrino (2010) use boosting with various base learners, e.g., regression trees among others, for one-period ahead forecasting of U.S. 3-month Treasury bill rates. The application of boosting by means of economic forecasting is the major novelty in the present work.

3.4.1 Forecasting Principles

Given that the set of observations $\mathbf{z}_t$ is called an information set, our objective is to use the information set to predict the unobserved outputs y_{t+h}. We use a direct forecasting strategy (e.g., Marcellino, Stock, and Watson, 2006; Chevillon and Hendry, 2005). The idea is to use a horizon-specific estimation model, where the response is the multi-period horizon. The direct forecasting approach is apparently a good choice under the presence of exogenous variables.

We need a cost function with which we can to evaluate the predictive accuracy. The choice of a cost function seems to be a large topic on its own. Hyndman and Koehler (2006) widely discussed and compared different measures of accuracy of times series forecasts. The references therein point the reader to different studies with often controversial conclusions about the “best” forecasting measure. Still, the literature being inconsistent, the MSE withstands the time proof and remains one

[1] Series USNA01.

of the most popular out-of-sample measures. Therefore, minimizing the quadratic expected cost

$$\mathrm{MSE} = \frac{1}{n} \sum_{t=T+1}^{T+n} \left[y_{t+h} - \hat{\mathrm{E}}\left(y_{t+h} | \mathbf{z}_t \right) \right]^2 \tag{3.12}$$

is set as the predictive accuracy measure. Expression (3.12) is known as the mean squared error, associated with the forecast $\hat{y}_{t+h} = \hat{\mathrm{E}}\left(y_{t+h} | \mathbf{z}_t \right)$ for horizon h and n forecasts in total.

Time series do not contain repeated measurements, therefore, a time series specific resampling scheme is needed in order to predict out-of-sample. One such scheme is the recursive scheme for forecasting. In that scheme, the starting point of the information set is fixed, usually at the beginning of the observed period. Then, we choose an initial time window for the information set and start to gradually increase its size. At the same time, we generate new forecasts for the unseen observations beyond that window. Therefore, our strategy is to combine the direct type of forecasting with a recursively enlarging information set.

3.4.2 Univariate Forecasting of Industrial Production

We apply GLMBoost, GAMBoost, BRUTO, and MARS to German industrial production. The univariate autoregressive model (AR) offers one of the simplest and most commonly used techniques for forecasting. It is easily applicable and therefore is often used as a benchmark model. The underlying assumption is that every alternative method should be at least as good as the autoregressive model in order to justify an increase in model's complexity.

The promising technique by Huang and Yang (2004) is omitted because Section 3.4.3 extends the available data set with exogenous variables, the so called leading indicators, and determines how the additional information affects the performance of the models. The inclusion of exogenous variables and their lags rapidly increases the number of covariates, forming a high-dimensional modeling problem. In this context, the method of Huang and Yang (2004) is no longer applicable.

We have a total number of 176 observations for IP. The initial information set is defined from the beginning 1992:01 until 2003:12, thus containing 144 observations. The maximum number of lags is limited to twelve. Therefore, the recursive scheme works as follows. At the first step twelve forecasts are calculated, i.e., prognoses for 2004:1-2004:12 are obtained. At the next step, the information set is enlarged by one and the horizons are re-estimated. We continue in this fashion until 2005:8 where

Table 3.3: Average squared forecast errors, multiplied by 10^3, of IP for 1, 6 and 12-periods ahead forecasts of the monthly industrial production growth rates in Germany. The results are based on 20 forecasts. Testing the null hypothesis that "the AR model is superior to the competing forecasting model" has been carried out by the Modified Diebold-Mariano Test (Harvey et al., 1997) and the results are represented by the p-values in parentheses.

Horizon	AR	GLMBoost	GAMBoost	BRUTO	MARS
1	.0668	.0648 (0.100)	.0698 (0.713)	.0704 (0.781)	.0916 (0.990)
6	.1052	.0808 (0.002)	.0848 (<0.001)	.1037 (0.357)	.0892 (<0.001)
12	.1214	.1220 (0.992)	.1093 (0.363)	.1161 (0.166)	.1014 (0.058)

the information set reaches its maximum. Thus, we complete twenty steps in total, i.e., $n = 20$ in (3.12).

Table 3.3 gives a summary of the average squared forecast errors for IP, obtained by the methods. It is apparent, that in short term forecasting the standard autoregressive model is quite a hard one to overcome. This simple, yet powerful, model is superior to BRUTO, MARS, and GAMBoost for short-term forecasting. On the other hand, GLMBoost seems to be more accurate in short term forecasting. With increasing forecasting horizon, all alternative models provide better forecasts for the monthly German industrial production growth rates, compared to AR. Both boosting methods prove to be efficient in forecasting, especially the linear boosting in short and middle-term forecasting, where it offers the smallest prediction error in average. For the longest horizon GLMBoost remains at least as good as AR, but performs relatively poorly in comparison to GAMBoost, BRUTO, and MARS. Figure 3.4 depicts the differences between the models of the prediction squared errors.

We employ the modified Diebold-Mariano test (Harvey et al., 1997) to check whether the outcome in Table 3.3 is due to chance. Harvey et al. (1997) proposed a finite sample correction of the original asymptotic test by Diebold and Mariano (1995a). We tested the null hypotheses "the AR model is superior to the competing forecasting strategy" in a series of pairwise comparisons with all models and the resulting p-values are shown in parentheses in Table 3.3. In addition, we report that both boosting techniques estimated quite large models (selected lags not shown), which is consistent with the results of the simulation study.

Based on the averaged errors in Table 3.3 and the given boxplots in Figure 3.4, it is rather challenging to announce a winning modeling strategy. It seems that the

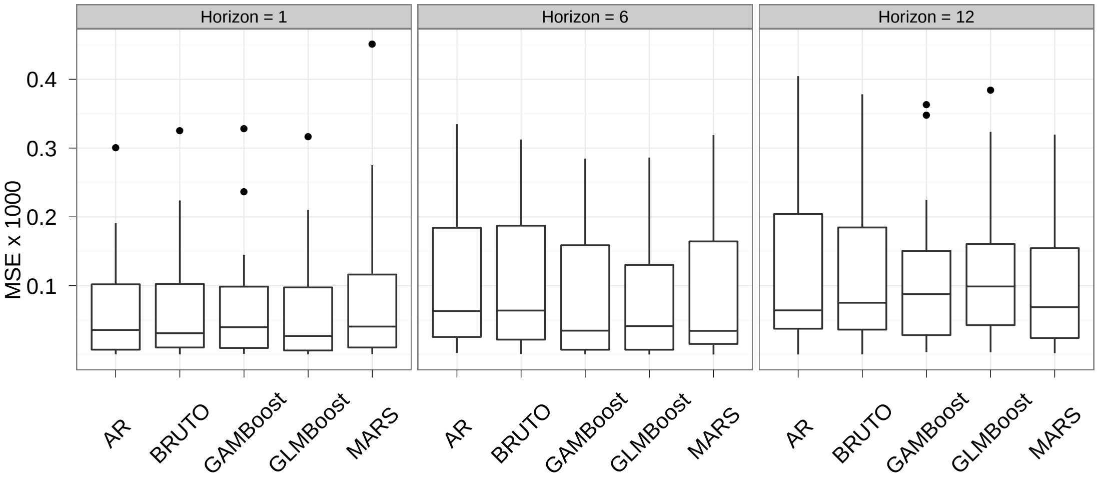


Figure 3.4: Boxplots of the average squared forecast errors (multiplied by 10^3) for 1, 6 and 12-periods ahead forecasts of the univariate IP, based on 20 forecasts.

models assimilate the information, based solely on IP, efficiently. Therefore, in order to improve the models prediction quality we supply them with additional information in the following section.

3.4.3 Forecasting Industrial Production with Exogenous Variables

Forecasting of industrial production is based on the assumption that different leading indicators should relate significantly with the response, and therefore positively influence its prediction. There are many leading indicators, however, that “claim” such an appealing property. Usually, one indicator is taken and its forecasting potential is judged by a bivariate autoregressive model, e.g., Dreger and Schumacher (2005) compared four indicators. The additional dimension does not necessarily improve the forecasting quality. On the contrary, in case of an “inappropriate” extra variable, it can even worsen the forecasting accuracy.

We collect the nine most commonly used indicators and investigate how they affect forecasting. The objective is to investigate if it is still possible to obtain good forecasts, despite the presence of probably redundant variables. Table 3.4 contains

Table 3.4: Leading Indicators.

Indicator	Provider	Label
Ifo Business Climate	Ifo Institute	ifo
ZEW Economic Sentiment	ZEW Institute	zew
OECD Composite leading indicator for Germany	OECD	oecd
Early Bird Indicator	Commerzbank	com
FAZ Indicator	FAZ Institute	faz
Interest Rate: overnight	IMF	rovnght
Interest Rate: spread	IMF	rspread
Employment Growth	Bundesbank	emp
Factor	Bundesbank	factor

a list of the nine frequently used leading indicators on forecasting German IP (see Appendix A for a detailed description of the indicators).

The vector autoregressive model has evolved as a standard tool in econometrics for analysing multivariate times series. Therefore, we will consider nine bivariate models, each consisting of the IP and one leading indicator from Table 3.4 in its restricted (VARr) and unrestricted (VAR) form. The restrictions are obtained via standard statistical t-tests.

The inclusion of one exogenous variable means that we fit a model with 24 covariates, i.e., twelve for the IP and twelve for the exogenous variable. The forecasting outcome is documented in Table 3.5. Every triplet shows the average performance of the corresponding models, respectively for 1, 6 and 12-periods ahead forecasts. In addition, it is indicated whether the forecast quality increased or decreased with respect to the univariate forecasts in Table 3.3. The change in the forecasting quality of both VAR and VARr is relative to the AR.

To allow for an easier comparison, Figure 3.5 visualizes Table 3.5; additionally the dashed red line shows the MSE of the AR model. Below, we give a summary of the empirical results:

(a) The out-of-sample forecasting results from Table 3.5 suggest that both boosting techniques remain robust to the impact of the exogenous variables. GLMBoost remains almost immune to redundant variables. Apparently, in five cases of middle to long-term forecasting (ifo, zew, oecd, faz and rovnght) GLMBoost did not consider the exogenous variable at all. This explains why these forecasts

Table 3.5: Average squared forecast errors of the monthly industrial production growth rates in Germany, with one leading indicator as an exogenous variable. The results are based on 20 forecasts, multiplied by 10^3. The symbol ▲indicates forecast improve with respect to Table 3.3 and ▽indicates decreased forecasting quality.

Indicator	Horizon	VAR	VARr	GLMBoost	GAMBoost	BRUTO	MARS
ifo	1	0.1101▽	0.0914▽	0.0647▲	0.0675▲	0.0845▽	0.0892▲
	6	0.1191▽	0.1291▽	0.0808▲	0.0826▲	0.1029▲	0.0899▽
	12	0.0947▲	0.1215▽	0.1220▲	0.1093▲	0.1168▽	0.1169▽
zew	1	0.0742▽	0.0724▽	0.0643▲	0.0754▽	0.0766▽	0.0826▲
	6	0.1116▽	0.1058▽	0.0808▲	0.0855▽	0.1157▽	0.0893▽
	12	0.0984▲	0.1151▲	0.1220▲	0.1076▲	0.1155▲	0.1164▽
oecd	1	0.0697▽	0.0697▽	0.0650▽	0.0727▽	0.0557▲	0.1041▽
	6	0.1055▽	0.1058▽	0.0808▲	0.0852▽	0.1245▽	0.0829▲
	12	0.1588▽	0.1141▲	0.1220▲	0.1100▽	0.1117▲	0.1188▽
com	1	0.0862▽	0.0840▽	0.0704▽	0.0751▽	0.0789▽	0.0764▲
	6	0.0981▲	0.0813▲	0.0803▲	0.0850▽	0.1093▽	0.0909▽
	12	0.1546▽	0.1163▲	0.1226▲	0.1093▲	0.1064▲	0.1069▽
faz	1	0.0698▽	0.0655▲	0.0648▲	0.0737▽	0.0830▽	0.0916▲
	6	0.3062▽	0.3203▽	0.0808▲	0.0848▲	0.1642▽	0.0895▽
	12	0.2156▽	0.1218▽	0.1220▲	0.1093▲	0.1389▽	0.1047▽
rovnght	1	0.0604▲	0.0605▲	0.0648▲	0.0731▽	0.0717▽	0.0910▲
	6	0.0958▽	0.1054▽	0.0808▲	0.0853▽	0.1111▽	0.0895▽
	12	0.1015▲	0.1151▲	0.1220▲	0.1093▲	0.1163▽	0.1017▽
rspread	1	0.0648▲	0.0581▲	0.0634▲	0.0701▽	0.0742▽	0.0927▽
	6	0.1010▽	0.1058▽	0.0808▲	0.0848▽	0.1005▲	0.0890▲
	12	0.1049▲	0.1150▲	0.1219▲	0.1093▲	0.1038▲	0.1052▽
emp	1	0.0671▽	0.0792▽	0.0632▲	0.0696▲	0.0704▲	0.0916▲
	6	0.0976▲	0.1004▲	0.1036▽	0.0946▽	0.1396▽	0.0919▽
	12	0.1090▲	0.1250▽	0.1356▽	0.1190▽	0.1361▽	0.1082▽
factor	1	0.0514▲	0.0519▲	0.0550▲	0.0684▲	0.0558▲	0.0948▽
	6	0.0988▲	0.1004▲	0.0861▽	0.0823▲	0.0990▽	0.0914▽
	12	0.1088▲	0.1077▲	0.1209▽	0.1161▽	0.1034▲	0.1147▽

are identical to the univariate case in Table 3.3. Transferred to the indicators, this interpretation suggests that they have only a short term effect on IP. In one-period ahead forecasting the exogenous variable exerted negative impact on GLMBoost in two cases only (zew, com) and outperformed AR in all cases except for the Early Bird indicator by the Commerzbank (com). In general, substantial changes of GLMBoost, compared to the univariate forecasting, were not found. That implies that linear boosting considered IP with its own lags to a larger extent than the remaining covariates. As a result, it showed a very strong overall performance and outperformed most of the models for one and six-periods ahead forecasts.

(b) The addition of exogenous variables changed the prediction power of GAMBoost, BRUTO, and MARS with varying success. Most notably GAMBoost and MARS show good and stable performance for six and twelve-periods ahead forecasts. This is best seen by the illustration in Figure 3.5. BRUTO improved its short term forecasting performance with almost every variable (except for the FAZ indicator), but in general remained worse than AR. In longer horizons, it showed a rather erratic behaviour.

(c) There are four leading indicators, which proved to have good forecasting quality in terms of bivariate linear autoregression. These are zew, faz, rspread, and factor which increased the forecasting precision of IP compared to AR. Moreover, the restricted bivariate autoregressive model with factor and faz provided the best short-term forecasts, but was easily outperformed for longer horizons. It is also evident that the restricted model is superior to the unrestricted one in most of the cases.

(d) From a computational point of view, MARS (2.3 sec.)[2], VAR (5.1 sec.) and VARr (9.8 sec.) were the fastest procedures. Closely followed by GLMBoost (17.5 sec.) and BRUTO (27.6 sec.) they all perform comparably fast. Boosting with P-spline weak learners (493.9 sec.) was more computationally demanding. It is probably worth nothing, that each additional covariate contributes to the boosting time linearly and, therefore, long computational times in boosting are inherited by computationally demanding base learners and should not be taken as evidence against the high-dimensional capabilities of boosting.

From the selection process (results are not shown) we gained and additional indication of the forecasting relevance of the indices. When selecting covariates, GLMBoost considered com, emp and factor more frequently than the others. Ifo, zew, oecd, and rspread seemed to have a short-term impact on the IP with their first lag being regularly selected. Still, IP with its own lags was dominant in the selection process. Boosting with P-spline weak learners was consistent with GLMBoost in terms of index-relevance but was prone to large models. BRUTO was the single

[2] User time measured on Linux, version 2.6.35-23-generic, 2 x Intel(R) Core(TM)2 Duo CPU T5750, 2.00GHz.

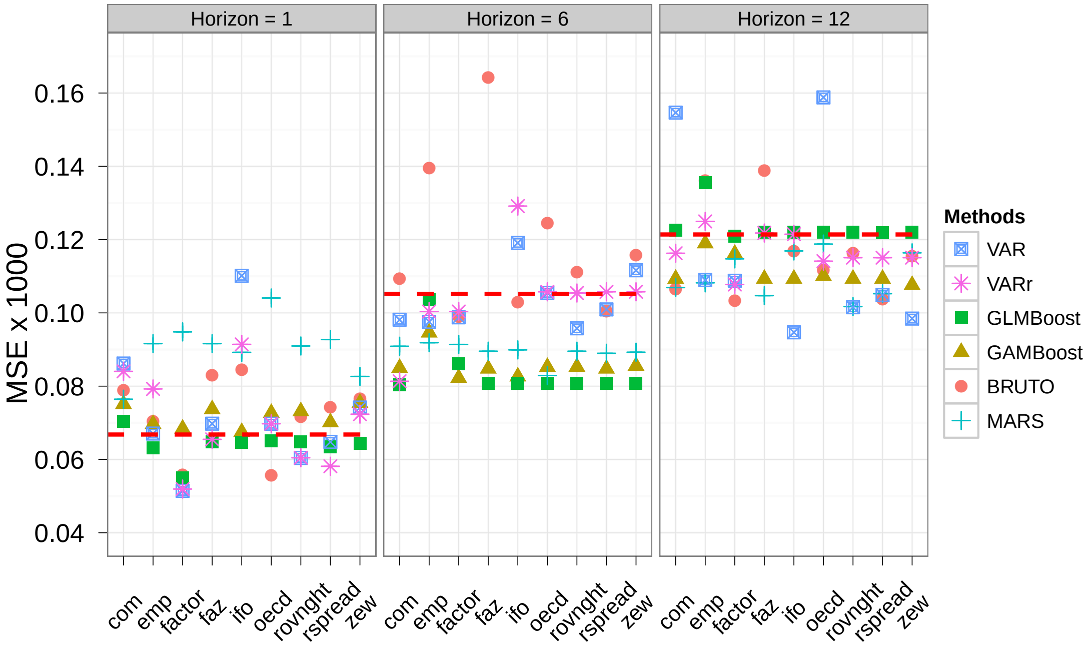


Figure 3.5: Average squared forecast errors of the monthly industrial production growth rates in Germany, with one leading indicator as an exogenous variable. Dashed red-line shows the value of the univariate autoregressive model. The results are based on 20 forecasts, multiplied by 10^3.

modeling strategy, which repeatedly considered more exogenous than endogenous lags. The indicators' dominance which occurred in this modeling strategy explains its erratic forecasting behaviour.

Again, testing whether the results in Figure 3.5 are due to chance is worth considering. The forecasting performance of AR was tested against each of the alternative methods. Rejecting the null hypothesis is interpreted as an evidence of the superior forecasting potential of the competing strategy. The results in terms of p-values are shown in Table 3.6. GLMBoost seems to be as good as the AR in the short-term forecasts, while being clearly superior in middle-term forecasts. Note that the equal p-value outcomes in the boosting strategies are due to the selecting property of boosting. A long-term forecasting horizon leads to a decrease in the information contributed by the exogenous variables and they are, therefore, disregarded by the selection mechanism.

Table 3.6: Pairwise comparisons of forecasts between the AR and the proposed. Testing the null hypothesis that "the AR model is superior to the competing forecasting model" has been carried out by the Modified Diebold-Mariano Test (Harvey et al., 1997) and the results are represented by the p-values.

Indicator	VAR	VARr	GLMBoost	GAMBoost	BRUTO	MARS
Horizon = 1						
ifo	0.989	0.921	0.101	0.598	0.877	0.956
zew	0.627	0.478	0.218	0.854	0.679	0.780
oecd	0.665	0.665	0.111	0.694	0.304	0.953
com	0.906	0.891	0.776	0.843	0.874	0.766
faz	0.579	0.410	0.101	0.721	0.772	0.991
rovnght	0.560	0.562	0.101	0.740	0.719	0.989
rspread	0.349	0.144	0.201	0.723	0.680	0.982
emp	0.097	0.980	0.071	0.713	0.781	0.991
factor	0.012	0.005	0.060	0.296	0.126	0.972
Horizon = 6						
ifo	0.578	0.917	0.002	< 0.001	0.478	0.001
zew	0.030	0.334	0.002	< 0.001	0.982	< 0.001
oecd	0.017	0.334	0.002	< 0.001	0.978	< 0.001
com	0.307	0.121	< 0.001	< 0.001	0.366	< 0.001
faz	0.849	0.825	0.002	< 0.001	1.000	< 0.001
rovnght	< 0.001	0.285	0.002	< 0.001	0.663	< 0.001
rspread	< 0.001	0.334	0.002	< 0.001	0.300	< 0.001
emp	0.276	0.207	0.336	0.031	0.954	< 0.001
factor	0.222	0.207	< 0.001	< 0.001	0.156	0.013
Horizon = 12						
ifo	0.172	0.694	0.992	0.363	0.458	0.769
zew	0.037	0.230	0.992	0.191	0.518	0.659
oecd	0.553	0.091	0.992	0.409	0.499	0.589
com	0.758	0.164	0.992	0.363	0.024	0.142
faz	0.864	0.591	0.992	0.363	0.729	0.163
rovnght	< 0.001	0.230	0.992	0.363	0.271	0.073
rspread	0.004	0.230	0.992	0.363	0.008	0.031
emp	0.294	0.356	0.993	0.777	0.535	0.095
factor	0.156	0.087	0.962	0.397	0.186	0.542

In conclusion, we found evidence that boosting can be very competitive in the forecasting of the industrial production in Germany. Particularly, boosting with linear base learners forecasts better than the linear autoregressive model. The increased flexibility of the nonparametric models does not seem to pay-off in short term foreacasting, but manages to improve the prediction quality when the information content decreases. This is typically observed in long-period forecasts. The endogenous lag effects had the biggest contribution to the forecasting quality, while the exogenous information affects essentially the short-term forecasts. In our analysis, the benchmark is the AR model and superiority is checked against this model. If one wants to select the best prediction model among all models the confidence set approach (Hansen, Lunde, and Nason, 2010) provides a strong tool.

3.5 Concluding Remarks

In this work, several parametric and nonparametric modeling techniques for autoregressive time series are compared, with particular focus on boosting methods. By letting the covariates be lagged values of a time series, we have applied various strategies to identify relevant lags, estimates, and forecasts. In Section 3.3 we proposed componentwise boosting of additive autoregressive model with P-spline weak learners. Alternative modeling strategies were also applied on several nonlinear autoregressive time series. It is evidenced that boosting of high-order autoregressive time series can be very competitive in terms of dynamics estimation. Unlike regression analysis, however, the serial dependence in time series data might mislead the fitting procedure to produce erroneous transformations. Care must be taken in using boosting algorithms in time series with strong serial correlation of the data. Further study on the use of boosting in time series context is needed to justify the general use of this procedure.

Another boosting strategy with parametric weak learners (GLMBoost) was included in order to perform a forecasting comparison, based on real world data in Section 3.4. The forecasting comparison was conducted over the monthly growth rates of German industrial production (IP). Both boosting strategies managed to outperform the benchmark in macroeconomic forecasting, namely the linear autoregressive model. Moreover, it became clear that GLMBoost was the most successful strategy in terms of short and middle-term forecasting.

Additionally, the model was extended with different exogenous variables (leading indicators). We had nine indicators available and we included each of them separately, in addition to the target variable, the industrial production. Our intention was to investigate whether these variables do indeed improve the forecasting quality of the industrial production and how boosting handles these high-dimensional models. Thus, having formed nine high-dimensional models, we forecasted the monthly growth rates of IP. Linear bivariate autoregressive models were also considered as standard tools for forecasting. Our approach, using componentwise linear and additive models in a function gradient descent algorithm, improves upon likelihood based boosting applied to nonlinear autoregressive times series models (Shafik and Tutz, 2009) in two respects. First, more flexible regression functions can be estimated using our approach (linear effects, decompositions of linear and smooth effects or interaction effects (Kneib et al., 2009)). Second, further research established alternative characteristics of the response to be regressed on lags or exogenous variables, most importantly quantile regression approaches implemented via componentwise functional gradient descent (Fenske et al., 2011).

The variables' impact on the forecasting quality had debatable success, since in many of the cases their inclusion worsened the forecasting performance, compared to the univariate case. GLMBoost, on the other hand, was almost immune to redundant variables by performing at least as good as in the univariate case. In one-period ahead forecasting, GAMBoost was affected by the additional variables rather strongly, which was counterproductive for its overall performance, when compared to the univariate case. The increased flexibility of GAMBoost was useful, however, in middle and long term forecasting, where the information content of the data is very low, i.e., it has low signal-to-noise ratio.

Another crucial topic for further development addresses the multivariate generalization of boosting. The first steps toward high dimensionality in the response were made by Lutz et al. (2008), who provided theoretical grounds and empirical evidence for its usability. Applying this approach would open new perspective for forecasting with boosting, based on iterative forecasts of multivariate models.

3.6 Computational Details

All data analyses presented in this work have been carried out using the R system for statistical computation (R Development Core Team, 2009). There are several implementations of boosting techniques, available as addon packages for R. Package *mboost* (Hothorn, Buehlmann, Kneib, Schmid, and Hofner, 2009) provides an implementation of gradient boosting with a large choice of base learners.

Our simulations were carried out with *mboost*. As weak learner we use P-splines, provided by the function `bbs()` and fitted by the `gamboost()` function. We use 20 knots (`knots = 20`), we set $M = 500$ (`mstop = 500`) as an upper bound for boosting and set the degrees of freedom to 3.5, i.e., `degree = 3.5`. The optimal number of steps is evaluated via the corrected AIC criterion provided by the `AIC()` function. For all other options we use the default values.

Further on, we considered the method proposed by Huang and Yang (2004), which uses spline fitting with BIC. Their approach was manually implemented since it is currently not available as an extension package for R or in any other statistical software. We used unpenalized cubic splines from *mgcv* package (Wood, 2006, 2009) to implement their method. The maximum number of candidate variables was equalled to the maximum number of lags.

An implementation of BRUTO can be found in package *mda* (Hastie, Tibshirani, Leisch, Hornik, and Ripley, 2009b). The corresponding function `bruto()` has a tuning parameter `cost` which specifies the cost per degree-of-freedom change. It was empirically investigated by Huang and Yang (2004) that a value of $\log(n)$ provides much better results than the default value of two, where n indicates the sample size. Therefore, in our application `cost` was set to $\log(n)$ too.

An implementation of MARS is available in package *mda* and the corresponding function is `mars()`. It has a tuning parameter which charges a cost per basis function, denoted by `penalty`. This tuning parameter was also set to $\log(n)$.

The estimation of AR is carried out via the `ar()` function in package *stats* with AIC criterion. The package *vars* (Pfaff, 2008) provides an implementation of the vector autoregressive model. We used a modified version of the function `VAR` in order to obtain direct forecasts.

Chapter 4

Boosting the Anatomy of Volatility

Financial risk, commonly represented by the volatility of asset prices, plays a major role in investment decisions. Therefore, understanding and predicting the volatility of financial instruments, asset classes, or financial markets in general, is of great importance for individual and institutional investors as well as financial regulators. In this work, based on Mittnik et al. (2012), we investigate a new strategy for understanding and predicting financial risk. We use componentwise gradient boosting techniques to identify the financial and macroeconomic factors that drive financial market risk and to assess the specific manner in which these factors affect future volatility. Componentwise boosting is a sequential learning method, which has the advantage that it can handle a large number of predictors and—in contrast to other machine learning techniques—gives rise to interpretable estimation results.

Adopting an EGARCH framework and employing a wide range of potential risk drivers, we derive monthly volatility predictions for stock, bond, commodity, and foreign exchange markets. Comparisons with alternative benchmark models show that these boosting techniques improve out-of-sample volatility forecasts, especially for medium- and long-run horizons. Moreover, we find that a number of risk drivers affect the volatility in a nonlinear fashion.

4.1 Introduction

The importance of understanding and adequately modeling financial market risk is widely recognized and has again become evident during the recent market turbulences. Volatility forecasts are used for risk management purposes, for example, to project risk measures, such as Value at Risk (VaR) and Expected Shortfall (ES), or to decide on hedging or other risk mitigation strategies.[1] They are also used for dynamic asset allocation decisions that are not just based on asset-specific risk but also on the dependence between assets, expressed in terms of time varying, volatility dependent measures, such as correlations or betas.

Although there has been a long history of efforts to predict asset returns (cf. Goyal and Welch, 2003; Welch and Goyal, 2008; Cochrane and Piazzesi, 2005; Lustig, Roussanov, and Verdelhan, 2011), the interest in volatility modeling started mostly with the seminal works of Engle (1982) and Bollerslev (1986) and has since become an intensely researched field in financial econometrics. However, only relatively few studies analyze the usefulness of financial and macroeconomic variables for volatility prediction. Schwert (1989) analyzes the relation of stock volatility and macroeconomic factors, such as GDP fluctuations, economic activity, and financial leverage, by employing autoregressive models. Engle, Ghysels, and Sohn (2008) use inflation and industrial production by combining a daily GARCH process with a mixed data sampling polynomial applied to monthly, quarterly, and bi-annual macroeconomic variables. Paye (2012) and, especially, Christiansen, Schmeling, and Schrimpf (2012) consider extended sets of macroeconomic factors as well as asset classes. Both use conventional linear models with log-transformed realized volatility as a normalized response and include lagged volatility, financial, and macroeconomic factors as regressors. Christiansen et al. (2012) employ Bayesian model averaging, but also restrict themselves to the family of linear models. In view of these competing approaches and given the range of alternative volatility concepts available, such as GARCH type, stochastic, implied, or realized volatility, it is no surprise that there is little or no general agreement on the application of financial and macroeconomic variables to volatility prediction.

The question of whether and, if so, how macro factors influence the volatility of asset prices is the focus of this work. To address this question, we use boosting techniques, a special machine learning method, to gain deeper insight into the nature of volatility processes. As will be shown, boosting techniques enable us not only to identify the factors driving market volatility, but also to assess the specific nature of their impact and, ultimately, help to improve prediction. Employing a broad set of potential macroeconomic and financial variables, we specify a flexible model, which is capable of capturing their—linear and nonlinear—influences on the volatility. In

[1] A comparison of alternative VaR forecasting strategies is given in Kuester, Mittnik, and Paolella (2006). For a discussion of the importance of volatility beyond economics, see Andersen, Bollerslev, Christoffersen, and Diebold (2006).

contrast to most of the existing literature, which focuses on stock market volatility (an exception is Christiansen et al., 2012), we analyze four diverse asset classes: stocks, bonds, commodities, and foreign exchange. We contribute to the existing literature on volatility modeling in several ways. We analyze the volatility of a range of relevant asset classes; we consider a broad set of possible macrodrivers; and, by employing boosting techniques, gain deeper insight into the nature of the forces driving asset price volatility.

In our analysis we use a version of the so-called componentwise, gradient boosting (see Bühlmann and Yu, 2003; Bühlmann and Hothorn, 2007a), which is designed to simultaneously select relevant factors and to model the specific nature of their impact. Boosting methods are especially suitable in applications where there are a large number of different but possibly "similar" predictors, as it handles multicollinearity problems by shrinking effects towards zero—a feature expected to be advantageous in out-of-sample predictions.

Volatility modeling with gradient boosting was first proposed by Audrino and Bühlmann (2003), who adopted a GARCH-type prediction model. They assume a stationary return process of the form $y_t = \sigma_t \varepsilon_t$, $\varepsilon_t \sim N(0,1)$, and a rather general dependence structure between σ_t and past returns. Their approach is, however, mainly suited for prediction, as it lacks any interpretability of the estimates. A fairly similar model with neural networks as base learners was proposed by Matías et al. (2010). Bühlmann and McNeil (2002) developed an alternative nonparametric first-order GARCH solution. They propose another strategy for GARCH(1,1) modeling which gives rise to interpretable estimates.

Although boosting has been proven to be a useful approach in many empirical applications, it has more or less been ignored in empirical economics or finance. Among the very few exceptions are Bai and Ng (2009), who use it for predictor selection and forecasting macroeconomic variables, and, as mentioned above, Audrino and Bühlmann (2009), who apply it to model the daily volatility of stock market indices. Our model differs from Audrino and Bühlmann (2009) in several respects, two of which we regard as particularly relevant. First, we go beyond the GARCH(1,1) specification by allowing both longer histories and exogenous factors to enter the model. The latter, as it turns out, clearly improves our understanding of volatility processes. Second, we employ componentwise predictor selection instead of the componentwise knot selection in tensor–spline estimation. This leads to genuinely different models and has the attractive feature that subjective decisions, such as the order of penalized B-splines, are avoided. Finally, given our goal to better understand the impact of macrofactors on volatility, we conduct our analyses at a monthly rather than daily frequency.

This chapter is organized as follows. Section 4.2 details and briefly illustrates the specific boosting algorithm we adopt. Section 4.3 describes the volatility measures and predictor variables used in the analysis and also the way the multistep forecasting comparisons are conducted. The empirical results for each of the four asset classes are presented in Section 4.4. Section 4.5 concludes.

4.2 Volatility Boosting Approach

In this section we give in detail the specific volatility boosting strategy used in the empirical application. We first present the underlying model specification and the particular boosting algorithm we adopt. Then we illustrate our approach using a small simulation study.

4.2.1 Proposed Model

Our volatility model corresponds to the exponential ARCH framework put forth by Nelson (1991), but allows the inclusion, in a rather flexible way, of risk drivers that can affect the volatility. In addition to a large number of drivers, we also include seasonal components, so that the total number of predictors is potentially very large and may even exceed the sample size. The proposed model is of the form

$$\begin{aligned} y_t &= \exp({}^{\eta_t}\!/_2)\varepsilon_t \\ \eta_t &= \beta_0 + f_{\text{time}}(t) + f_{\text{year}}(n_t) + f_{\text{month}}(m_t) + \sum_{j=1}^{s} f_j(y_{t-j}) + \sum_{k=1}^{q}\sum_{j=1}^{p} f_j^{(k)}(x_{t-j}^{(k)}) \qquad (4.1) \\ &= \eta(\mathbf{z}_t), \end{aligned}$$

where y_t denotes logarithmic returns, i.e., $y_t = \log(P_t/P_{t-1})$, with $P_1, \ldots, P_T$ denoting the observed asset prices, and $\varepsilon_t \sim N(0,1)$. The r dimensional vector $\mathbf{z}_t = (1, t, n_t, m_t, y_{t-1}, \ldots, y_{t-s}, x_{t-1}^{(1)}, \ldots, x_{t-p}^{(1)}, \ldots, x_{t-1}^{(q)}, \ldots, x_{t-p}^{(q)})^\top$, with $r = s + qp + 4$, contains the predictor realizations available at or prior to time $t-1$. The function $f_{\text{month}}(m_t)$, $m_t \in \{1, 2, \ldots, 12\}$ captures the possible deterministic seasonal patterns in the volatility; $f_{\text{year}}(n_t)$, n_t describes the typical annual fluctuations, which occur throughout the sample period; $f_{\text{time}}(t)$, $t \in \{1, \ldots, T\}$, models the volatility trend; $f_j(y_{t-j}), j = 1, \ldots, s$, capture the influence of past returns; and $f_j^{(k)}(x_{t-j}^{(k)}), j = 1, \ldots, p$ are functions of the lagged factor $k \in \{1, \ldots, q\}$.

All the $f_{\cdot}(.)$ functions in (4.1) are specified as regression trees. Regression trees are a nonparametric technique that can handle complex and abruptly varying forms of dependence by recursively partitioning the predictor domain into groups with similar response values and assigning a constant value to the response within each group.[2] Specifically, we use *conditional inference trees* (Hothorn, Hornik, and Zeileis, 2006). Therefore, the model can be interpreted as a regime-dependent volatility–response model, which partitions the predictor space according to the magnitude with which the conditional volatility responds. Both linear estimation and nonparametric, smooth estimation of $f_{\cdot}(.)$, as well as a combination of the two, can be specified.[3]

We estimate (4.1) via componentwise, gradient boosting, which derives the final model by sequentially combining a series of individual predictor components. To avoid overfitting in the first step, we control the bias variance tradeoff by using a low-variance/high-bias model. In subsequent steps this bias will be iteratively reduced, with the variance increasing at a slower rate (Bühlmann and Yu, 2003). Our estimation minimizes the expectation of some loss function, L, such that

$$\hat{\eta} = \arg\min_{\eta} \mathbb{E} L(y_t, \eta(\mathbf{z}_t)) \tag{4.2}$$

and $\exp\left(\hat{\eta}(\mathbf{z}_t)\right) = \mathbb{V}\left(y_t|\mathbf{z}_t\right)$, with L being differentiable with respect to η. To obtain a solution in the data rather than function space, we parameterize η by

$$\hat{\eta} = \arg\min_{\eta} \frac{1}{T} \sum_{t=1}^{T} L(y_t, \eta(\mathbf{z}_t; \boldsymbol{\beta})). \tag{4.3}$$

The solution to (4.3) is derived by reducing the empirical loss in successive steps as described in Chapter 2. To estimate the desired characteristic of the conditional distribution (here, the conditional variance), the loss function, L, needs to be appropriately specified. We do so by assuming $y_t|z_t \sim N(0, e^{\eta_t})$, so that the negative conditional log-likelihood function is the empirical loss function, i.e.,

$$L_t = \frac{1}{2}\left[\eta_t + \frac{y_t^2}{e^{\eta_t}}\right], \tag{4.4}$$

giving rise to the negative gradient

$$g_t = -\frac{\partial L_t}{\partial \eta_t} = \frac{1}{2}\left[\frac{y_t^2}{e^{\eta_t}} - 1\right]. \tag{4.5}$$

[2] For a detailed treatment of the algorithms behind regression trees, see Breiman et al. (1984).

[3] For an implementation, we refer to the R addon package *mboost* (R Development Core Team, 2012; Hothorn, Buehlmann, Kneib, Schmid, and Hofner, 2011).

As explained in Chapter 2, boosting favors the direction given by the largest reduction in the empirical loss, i.e., the direction specified by the negative gradient. This means that we seek the solution in the data space by fitting the covariates against the negative gradient.

Instead of jointly fitting all covariates, they are fitted individually through base learners. Therefore, we get r individual models for the covariates. As individual models, we choose the conditional inference regression trees (Hothorn et al., 2006) with two nodes, also called "stumps." Modeling the dependence between the response and the covariate in terms of two constants assigned to disjoint groups is naturally inflexible and cannot fit the complete signal in a single step. This bias is reduced due to the iterative nature of the algorithm, which slowly adapts to the underlying signal.

In addition, we shrink the coefficient towards zero, as proposed by Friedman (2001). Shrinkage helps to dampen the "greediness" of the gradient technique, which may otherwise be prone to neglect correlated predictor candidates, and "cures" the typical instability of forward selection methods (Breiman, 1996). The "right" amount of shrinkage is determined empirically and can safely vary between 1% and 10%. The specific choice mainly affects the computational time only. Fitting the base learner will modify evaluation of the gradient in the next step, and, with each step, the covariates and gradients become more and more orthogonal.

Note that we can choose any statistical model for the base learners. In our applications, a specification via stumps turned out to be a better choice than, for example, smooth P-splines or a simple linear model. This seems largely due to the abrupt changes we observe in the volatility.

Without stopping, boosting with stumps will inevitably overfit and ultimately lead to a perfect fit, making the model useless for prediction. Therefore, an appropriate stopping rule is essential. The optimal number of boosting steps can be determined by bootstrapping, where we sample (with replacement) from the data with probability $1/T$ as if they originated from a multinomial distribution. Thus, each sample uses roughly 64% of the original data for training and the remaining, unselected, data points are used for evaluation. We repeat this twenty-five times for a large number of boosting steps and choose the step number that produces the lowest average out-of-sample loss.

To summarize, the boosting algorithm which we employ consists of the following steps:

1. Initialize the function estimate $\hat{\eta}_t^{[0]} = \log\left(\frac{1}{T-1}\sum_{t=1}^{T}(y_t - \bar{y})^2\right), \bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t,$

$t = 1, \dots, T$.

2. Specify the set of base learners in terms of regression trees: $f_{\cdot}(z_t) = \sum_{j=1}^{J} \gamma_j I_{R_j}(z_t)$, $\forall z_t \in \mathbf{z}_t$. We use stumps, so each tree has only $J = 2$ leaves. Denote the number of base learners by r and set $m = 0$.

3. Increase m by one.

4. (a) Compute the negative gradient (5.10) and evaluate $\hat{\eta}^{[m-1]}(\mathbf{z}_t)$, $t = 1, \dots, T$.
   (b) Estimate the negative gradient, using the stumps specified in Step 2. This yields r vectors, where each vector is an estimate of the gradient.
   (c) Select the base learner $\hat{f}^{[m]}$ that correlates most with the gradient according to the residual sum of squares criterion.
   (d) Update the current estimate by setting $\hat{\eta}^{[m]} = \hat{\eta}^{[m-1]} + \nu \hat{f}^{[m]}$, where ν is regarded as a *shrinkage* parameter or as a step size.

5. Repeat Steps 3 and 4 until the stopping condition applies.

4.2.2 An Illustration

To illustrate our volatility boosting approach, we run a small simulation using the data generating process

$$\begin{aligned} y_t &= \exp({}^{\eta_t}\!/_2)\varepsilon_t \\ \eta_t &= 0.1 + 2 \cdot x_{t-1}^{(1)} + 2 \cdot \mathrm{I}_{[0.1,0.5]}(x_{t-1}^{(2)}) \cdot x_{t-1}^{(2)} - 0.6 \cdot \mathrm{I}_{[-0.5,-0.2]}(x_{t-1}^{(3)}) + \\ &\quad 0 \cdot x_{t-1}^{(4)} + 0 \cdot x_{t-1}^{(5)} + 0 \cdot x_{t-1}^{(6)}, \end{aligned} \tag{4.6}$$

with $\varepsilon_t \sim N(0,1)$, and $x_{t-1}^{(i)}$ being the $(t-1)$-th observation of $X_i \sim U[-0.5, 0.5]$, $i = 1, \dots, 6$, $t = 1, \dots, T$, with $T = 400$ and $\mathrm{I}_A(\cdot)$ denotes the indicator function, such that $\mathrm{I}_A(x) = 1$, if $x \in A \subset \mathbb{R}$, and $\mathrm{I}_A(x) = 0$, otherwise. Note that only the first three covariates contribute to the volatility—the first linearly, the second linearly only for $X_2 \in [0.1, 0.5]$, the third in the form of a step function. The last three covariates, X_4 through X_6, do not contribute, and are included for checking the robustness against false detection. We choose linear base learners for all but the second and third predictors, which are fitted with regression-tree base learners. The model for boosting is then given by

$$y_t = \exp(\eta_t/2)\varepsilon_t$$
$$\eta_t = \beta_0 + \beta_1 x_{t-1}^{(1)} + \sum_{j=1}^{J_1} \gamma_j^{(2)} \mathrm{I}_{R_j^{(2)}}(x_{t-1}^{(2)}) + \sum_{j=1}^{J_2} \gamma_j^{(3)} \mathrm{I}_{R_j^{(3)}}(x_{t-1}^{(2)})) + \beta_4 x_{t-1}^{(4)} + \beta_5 x_{t-1}^{(5)} + \beta_6 x_{t-1}^{(6)}, \tag{4.7}$$

where $R_j^{(2)}$ and $R_j^{(3)}$ denote the estimated partitions in the domain of X_2 and X_3. The splitting decisions are made by using the permutation test (Strasser and Weber, 1999), which measures the level of dependence between the gradient and the corresponding covariate. Its test statistic is maximized among all possible split positions (see also Hothorn et al., 2006).

Ideally, the algorithm will recover the β and $\gamma^{(3)}$ parameter values specified in (4.7). This means that X_4, X_5 and X_6 should not be selected, i.e., $\beta_4 = \beta_5 = \beta_6 = 0$, and that the domain of X_3 should be partitioned into the defined regions with only interval $X_3 \in [-0.5, -0.2]$ affecting volatility. Regarding X_2, although having a linear form $X_2 \in [0.1, 0.5]$ and zero impact otherwise, we intentionally chose an "incorrect" base learner, namely a step function, to see whether the influences can still be adequately approximated.

Figure 4.1 shows the simulated, driver-specific return components (upper panel) and the estimated partial volatility impacts in a log scale (lower panel). The influence of the underlying volatility drivers turns out to be captured reasonably well. The parameter estimate $\hat{\beta}_1 = 1.463$ is underestimated due to parameter regularization via early stopping. This is typical for shrinkage methods in finite samples, where the parameter estimates usually have smaller magnitudes than the unregularized solutions, and the bias vanishes as the sample size increases. The advantage of early stopping is that the redundant predictors are never selected, i.e., $\hat{\beta}_4 = \hat{\beta}_5 = \hat{\beta}_6 = 0$. Furthermore, X_3 has the largest jumps near the right border of the interval $[-0.5, -0.2]$, and the linear structure of $X_2 \in [0.2, 0.5]$ is also captured reasonably well despite the moderate sample size chosen for purposes of illustration.

The results shown in Figure 4.1 are typical in the sense that the deviations of several hundred repetitions were small. If we translate the log scale from Figure 4.1 into the standard deviation, we obtain an estimate of the whole conditional density. Figure 4.2 (upper panel) shows the estimated partial densities for the first three covariates with the central 95% interquantile range (darker color). Figure 4.2 (lower panel) also shows the empirical conditional density for simulated return observations. Visual inspection reveals that the variation in the volatility is closely captured, a

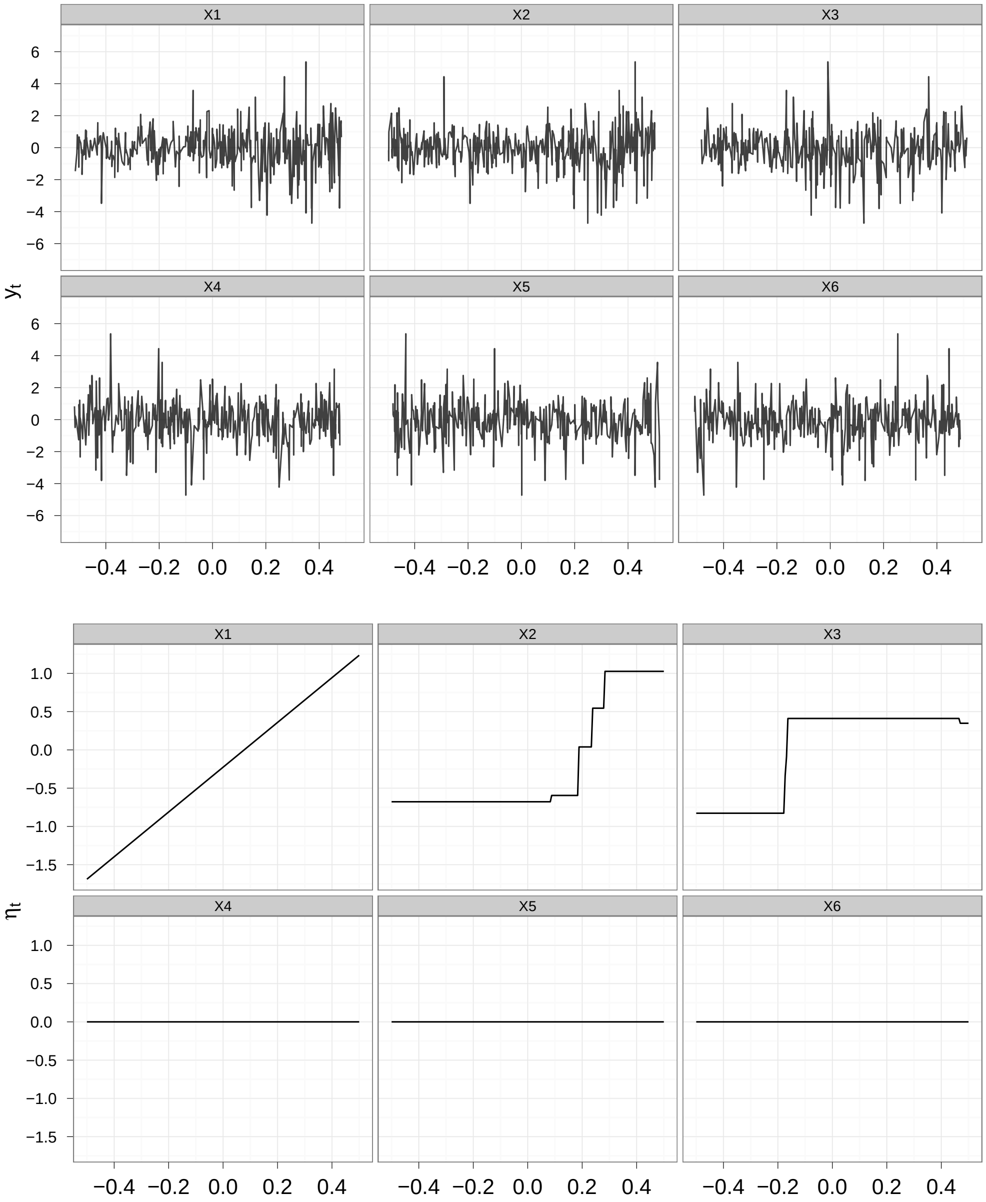


Figure 4.1: Partial returns (upper panel) simulated from (4.6) indicate how they are affected by drivers X_1 through X_6. Estimated partial volatility (lower panel) for model (4.6). The volatility η_t is measured on the log scale.

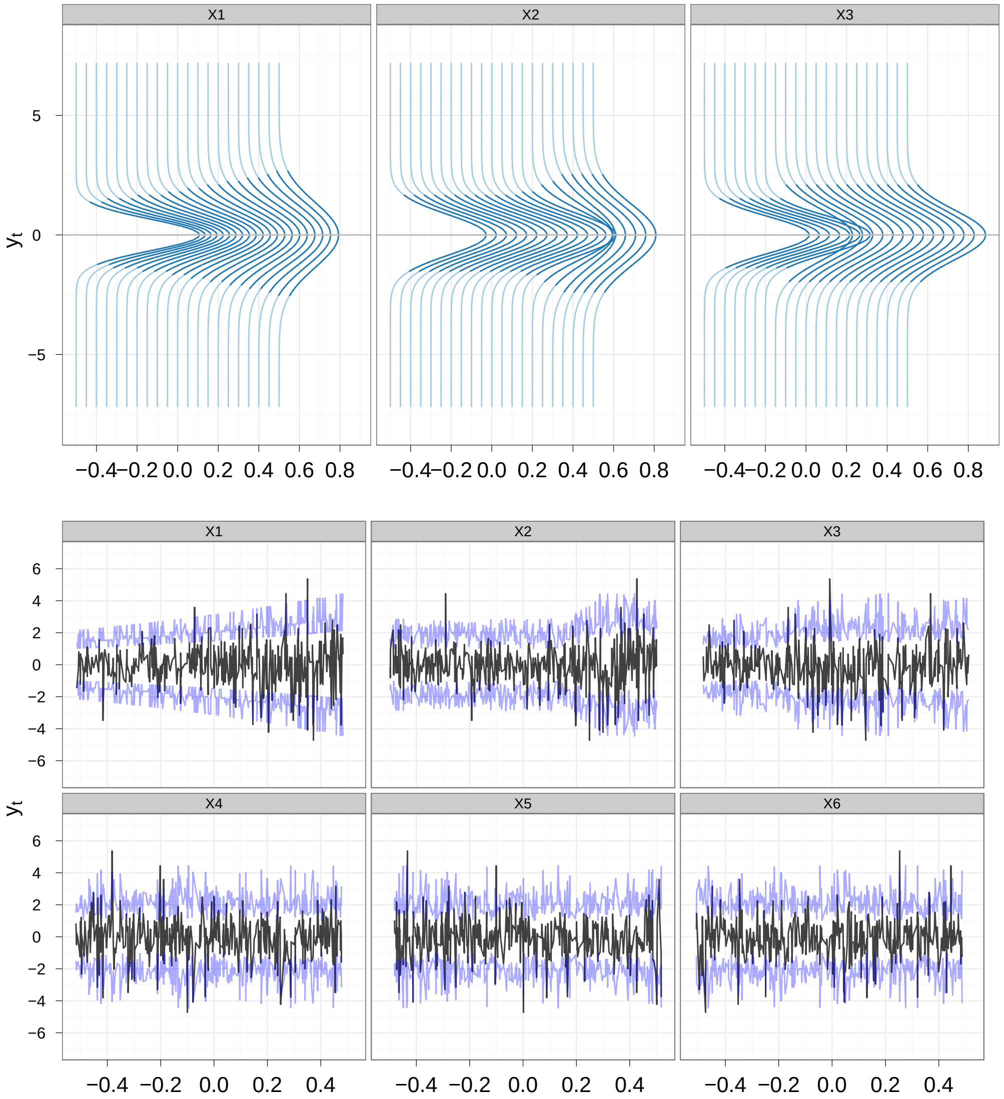


Figure 4.2: Partial conditional density estimation (upper panel) affected by X_1 through X_3 in model (4.6). The dark lines indicate the estimated 95% interquantile range, the lighter ones show the estimated tails. Partial returns (lower panel, black lines) indicate how they are affected by drivers X_1 through X_6. The blue lines represent the 95% interquantile range of the conditional density.

finding that is confirmed by the fact that estimates produce a coverage rate of 95.75% for the 95% interquantile range. The partial contribution of each covariate is readily obtained in an interpretable way: an increase in X_1 causes larger variance; X_2 is positively correlated with the variance for $X_2 \in [0.2, 0.5]$; the variance contribution markedly decreases for $X_3 \in [-0.5, -0.2]$; all other components have no effect, so that the conditional density remains invariant with respect to X_4, X_5 or X_6.

By providing such detailed and interpretable insight into the the nature of volatility processes, the volatility boosting strategy proposed here should help to improve our understanding about the risk drivers in financial markets. To what extent this insight translates into better risk predictions in practice is the focus of the next section.

4.3 An Empirical Application to Four Asset Classes

In this section, we present an empirical application of our approach to volatility prediction considering four diverse asset classes. First, we briefly describe the data employed, i.e., the data for the assets to be modeled as well as the financial and macroeconomic factors entertained as the potential volatility drivers. Then, we will discuss the procedure we use to evaluate the predictive performance.

4.3.1 The Data

We investigate the predictability of volatility of four asset types, namely, stocks, bonds, commodities, and foreign exchange, for each of which we select a representative index. The equity market is represented by a S&P 500 futures contract traded on the Chicago Mercantile Exchange; for the bond market, we use 10-year treasury note futures contracts traded on the Chicago Board of Trade; the commodity market is represented by Standard & Poor's GSCI commodity index; and we use a trade-weighted currency portfolio provided by the Federal Reserve Bank of St. Louis to proxy foreign currency investments. The latter is a weighted average of the foreign exchange value of the U.S. dollar against a broad set of currencies that circulate widely outside their countries of issue, including the Euro Area, Canada, Japan, the United Kingdom, Switzerland, Australia, and Sweden. The data set covers the period from February 1983 to September 2010 and consists of 332 months in total. Various summary statistics for the four return series and the logarithmic realized volatility. Series are given in Tables 4.1 and 4.2, respectively.

As potential volatility drivers over that period, we consider a fairly exhaustive set of 26 financial and macroeconomic factors, which are listed in Table 4.3. It

Table 4.1: Descriptive statistics for the return series.

	Mean	Std Dev	Skewness	Kurtosis	AR1
Stock	0.0062	0.0457	-1.0116	5.8971	0.0661
Commodity	0.0057	0.0578	-0.6082	6.6808	0.1958
Bond	0.0016	0.0204	0.0504	3.8869	0.0501
FX	-0.0015	0.0213	0.0734	3.5620	0.0743

Table 4.2: Descriptive statistics for the log realized volatility series.

	Mean	Std Dev	Skewness	Kurtosis	AR1
Stock	-6.3278	0.9254	0.8149	2.0479	0.6669
Commodity	-6.1654	0.9327	0.3341	0.0438	0.7797
Bond	-8.0330	0.7128	-0.0012	0.0987	0.5807
FX	-8.0280	0.6943	0.0503	0.4122	0.5613

includes the explanatory variables (resp. transformations thereof) used by Welch and Goyal (2008) for predicting stock market returns, namely, book to market ratio, net equity expansion, term spread, relative T-Bill rate, relative bond rate, long-term bond return, and default spread (see Table 4.3 for more details). In addition, we include the three Fama–French factors: the U.S. market excess return, the size, and the value factor.

The set of predictors also contains the Pastor and Stambaugh (2003) liquidity factor, the return on the MSCI world index, the TED spread (i.e., the difference between the three-month LIBOR rate and the T-Bill rate), the Cochrane and Piazzesi (2005) bond factor, the return on the CRB spot index, the carry trade factor as in Lustig et al. (2011), the return on dollar risk factor introduced by Lustig et al. (2011), and the FX average bid–ask spread (Menkhoff et al., 2011).

In addition, the set of potential drivers includes various macroeconomic variables: M1 growth, the purchasing manager index, housing starts, inflation, U.S. industrial production growth, and new orders of consumer goods and materials. Finally, we also consider the Financial Stability Index (FSI) for the U.S., which was developed by the International Monetary Fund (Cardarelli, Elekdag, and Lall, 2009).

Table 4.3: Description of the predictor variables employed.

Variable	Abbrev.	Source	Description
U.S. Market Excess Return	MKTRF	Homepage FF	The Fama-French market factor: U.S. stock return minus one-month T-Bill rate
Size factor	SMB	Homepage FF	The Fama-French's SMB factor: return on value small minus return on big stocks
Value factor	HML	Homepage FF	The Fama-French HML factor: return on value stocks minus return on growth stocks
U.S. IP growth	IPGR	Datastream	Log of annual growth rate of U.S. industrial production
New orders consumer growth	ORDc	Datastream	New orders of consumer goods and materials; log of annual growth rate
U.S. M1 growth	M1c	Datastream	Log of annual growth rate of U.S. M1
Consumer Sentiment index	SENTc	Datastream	Monthly change in consumer sentiment
Purchasing Manager index	PMIc	Datastream	Monthly change in purchasing manager index
Housing Starts	HSc	Datastream	Monthly change in housing starts
TED spread	TED	Datastream	Measure of illiquidity: LIBOR minus T-Bill rate
MSCI World	MSCIc	Datastream	Return on the MSCI world stock market index
CRB Spot Index annual ret	CRBc	Datastream	Measure of growth in commodity prices: annual log difference of CRB spot index
Book to market ratio	b/m	Goyal-Welch	Ratio of book value to market value for the Dow Jones Industrial Average
Net equity expansion	ntis	Goyal-Welch	Ratio of 12-month moving sums of net issues by NYSE listed stocks divided by total NYSE market capitalization (end of year)
Inflation	infl	Goyal-Welch	Annual growth rate of the U.S. consumer price index
Long term rate of return	ltr	Goyal-Welch	Rate of return on long term government bonds
Default spread	DEF	Goyal-Welch	Measure of default risk of corporate bonds: BAA bond yields minus AAA bond yields
Relative T-Bill rate	RTB	Goyal-Welch	T-Bill rate minus its 12 month moving average
Relative bond rate	RBR	Goyal-Welch	Long-term bond yield minus its 12 month moving average
Term spread	TS	Goyal-Welch	Long-term bond yield minus three-month T-Bill rate
Pastor-Stambaugh liquidity factor	liq_fac	Pastor and Stambaugh (2003)	Measure of stock market liquidity based on price reversals Pastor and Stambaugh (2003)
FX average bid-ask spread	BAS	Menkhoff, Sarno, Schmeling, and Schrimpf (2011)	Measure of illiquidity in the foreign exchange market calculated from bid-ask spreads
Return on Dollar risk factor	RX	Lustig et al. (2011)	FX risk premium measure: average premium for bearing FX risk
Carry trade factor	HML	Lustig et al. (2011)	Return on high interest rate currencies minus return on low interest currencies
Cochrance Piazessi factor	CP		Measure of bond risk premia; recursively estimated based on Fama-Bliss file according to Cochrane and Piazzesi (2005)
Financial Stress Index	FSI	IMF	The FSI for the U.S. comprises seven variables which serve to capture three financial market segments.

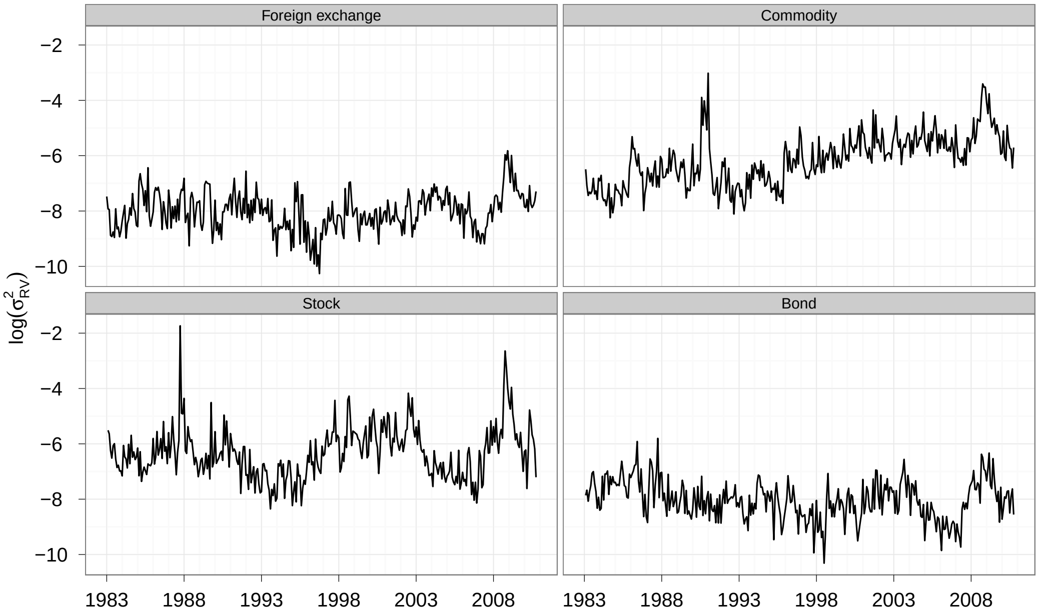


Figure 4.3: Time series plots of the monthly realized volatility (in logarithms) as defined in Equation (4.8).

4.3.2 Analyzing the Predictive Performance

Volatility is inherently unobservable, so that measuring volatility is a challenge. In this paper, we follow the tradition of French, Schwert, and Stambaugh (1987) and Schwert (1989) and use monthly *realized volatility*, calculated from daily returns, as proxy for volatility, and for evaluating the predictive performance of our volatility models.[4] The realized volatility for asset i in month t, denoted by $RV_{i,t}$, is defined by

$$RV_{i,t} = \log \sum_{\tau=1}^{M_t} r^2_{i,t,\tau}, \quad t = 1, \ldots, T, \tag{4.8}$$

where $r_{i,t,\tau}$ denotes the τth daily return of asset i in month t; and M_t is the number of trading days in month t. Figure 4.3 shows the resulting realized volatility time series for the asset under investigation.

The predictive performance is examined over the period June 2002 to September 2010. We use a rolling window scheme for forecasting. Starting with a history of 230

[4] For a review of the realized–volatility concept, we refer to Andersen et al. (2006).

months, we move the fixed-length window forward month by month, re-estimate, and, for each asset class, generate a sequence of one-step-ahead forecasts over a period of 100 months.[5] Applying a direct forecasting approach,[6] we also produce multi-step forecasts for horizons of up to six months.

We include the first and second lag of all 26 factors as predictors, so that, in (4.1), $q = 26$ and $p = 2$. In addition, we include lags one and two of the realized volatility ($s = 2$), to capture the state dependence and any autoregressive behavior in volatility. Allowing also for seasonal components, we have a total of $r = 58$ predictors.

As volatility is latent, it is common to use the squared returns y_t^2 as a proxy. However, as this estimator is very noisy, we follow another approach. We evaluate the forecasting performance in terms of the mean squared error between the "true" (realized) volatility, as defined in (4.8), and our forecasts for η_{t+h}. Doing so, the h-step squared prediction error for asset i is given by[7]

$$\mathrm{ERR}_{t+h} = (RV_{i,t+h} - \eta_{t+h})^2. \tag{4.9}$$

We derive direct h-step forecasts by adapting (4.1) to the forecasting horizon of interest, i.e.,

$$\begin{aligned} y_{t+h} &= \exp(\eta_{t+h}/2)\varepsilon_{t+h}, \text{ for } h = 1, \dots, 6, \\ \eta_{t+h} &= \beta_0 + f_{\text{time}}(t+h) + f_{\text{year}}(n_{t+h}) + f_{\text{month}}(m_{t+h}) + \sum_{j=0}^{s-1} f_j(y_{t-j}) + \sum_{k=1}^{q}\sum_{j=0}^{p-1} f_{k,j}(x_{k,t-j}) \\ &= \eta_h(\mathbf{z}_t). \end{aligned} \tag{4.10}$$

Next, the empirical results will be discussed.

4.4 Empirical Results

In discussing the empirical results we focus on the questions which factors drive realized volatility and to what extend they do so. One finding is that the driving

[5] Two observations are "lost" due to lagged variables.

[6] For direct forecasting via boosting in a nonlinear time series context, see Robinzonov et al. (2012).

[7] For a detailed comparison and discussion of different forecast evaluation criteria for realized volatility see Patton (2011).

Table 4.4: Out-of-sample forecast evaluation.

	Theil's U				Out-of-sample R^2			
Hor.	Stock	Commodity	Bond	FX	Stock	Commodity	Bond	FX
1	0.99	1.07	1.04	1.03	0.00	-0.15	-0.09	-0.06
2	0.98	0.86	1.01	1.01	0.02	0.25	-0.03	-0.02
3	0.89	0.92	1.00	0.91	0.20	0.14	0.00	0.15
4	0.85	0.90	0.98	0.93	0.27	0.18	0.03	0.13
5	0.87	0.79	0.94	0.96	0.24	0.37	0.11	0.06
6	0.83	0.71	0.88	0.95	0.30	0.49	0.21	0.08

factors exert a highly nonlinear influence on volatility. This is evident when comparing the forecasting performance based on linear base learners to that derived from (nonlinear) regression trees, as linear base learners give rise to a lower forecasting accuracy.

To assess the predictive performance, we compare multi–step, out–of–sample forecasts from the proposed boosting procedure to those of a GARCH(1,1) benchmark model.[8] Clearly, there are many potential alternatives that could serve as benchmarks.[9] However, in the spirit of the article "A forecast comparison of volatility models: does anything beat a GARCH(1,1)?" by Lunde and Hansen (2005), the GARCH(1,1) model can be regarded as a natural and challenging benchmark model in this context.

In the following subsections, we evaluate the forecast performance of the boosting approach and discuss in some detail the driving factors of each market.

4.4.1 Forecast Evaluation

To evaluate the out-of-sample forecasts, we compute Theil's U and out-of-sample R^2 statistics for horizons ranging from one to six months. Theil's U is defined as the ratio of the root mean squared error (RMSE) of our model and to that of the benchmark model. A value smaller than unity indicates that our model outperforms the benchmark model in terms of forecasting accuracy. The out-of-sample R^2, proposed

[8] The multi–step GARCH forecasts are made recursively. We also carried out direct h–step ahead GARCH forecasts by estimating GARCH models for each corresponding frequency. However, these noniterative forecasts performed rather poorly.

[9] Christiansen et al. (2012) use, for example, an autoregressive model for realized volatility as a benchmark.

Table 4.5: Modified Diebold–Mariano test results. *,**,*** denote significance at 10%, 5%, and 1%, respectively.

Horizon	Stock	Commodity	Bond	FX
1	0.490	0.790	0.858	0.630
2	0.441	0.011 **	0.633	0.555
3	0.104	0.114	0.501	0.215
4	0.052 *	0.071 *	0.357	0.288
5	0.050 **	0.016 **	0.154	0.370
6	0.011 **	0.009 ***	0.030 **	0.355

by Campbell and Thompson (2008) has an interpretation that is similar to that of Theil's U. Letting in market i, $\eta^M_{i,t+1}$ and $\eta^B_{i,t+1}$ denote the forecasts from our model and those of the benchmark, respectively, the out-of-sample R^2 is defined by

$$R^2_{OOS} = 1 - \frac{\sum_{t=R}^{T-1} \left(RV_{i,t+1} - \eta^M_{i,t+1}\right)^2}{\sum_{t=R}^{T-1} \left(RV_{i,t+1} - \eta^B_{i,t+1}\right)^2}, \tag{4.11}$$

where T denotes the total sample size, and R the initialization period. Positive (negative) values of R^2_{OOS} indicate that the boosting approach provides a superior (inferior) forecasting accuracy relative to the benchmark.

The estimation results for all markets are shown in Table 4.4. For stock–market volatility, we find that the boosting approach outperforms the benchmark over all horizons. For the other markets, the benchmark produces better one–step and in case of bonds and foreign exchange also better two–step predictions. In all other cases, especially for predictions beyond two months, the boosting approach dominates. For commodities and stocks, and to a lesser extend, for bonds, the medium-term performance is considerably better, whereas for FX volatility the difference seems to be negligible.

Finally, we apply the Diebold–Mariano test (Diebold and Mariano, 1995b) in the modified version of Harvey et al. (1997) to assess forecasting accuracy. The null hypothesis of the test is that the benchmark model's forecasting error is smaller than that of the proposed model. Therefore, rejection of the null hypothesis favors our approach. Table 4.5 reports the p-values of the modified Diebold–Mariano test for all six forecasting horizons. The results are in line with those indicated by Theil's U and the out-of-sample R^2 statistics. Inclusion of exogenous factors as well as the regime–dependent estimation in the boosting approach help to improve medium–

and long–term volatility forecasting—especially for commodity and stock markets.[10] Overall, the forecasting comparisons suggest that boosting leads to short–term forecasts that are of similar quality as those of a GARCH(1,1) model but considerably more accurate in the medium– and long–term. Here, an important observation is that, in general, the GARCH forecasts have a wider MSE range measured by their central 75%–quantiles. Thus, it seems that boosting delievers more robust forecasts. Boxplots of the mean squared errors (MSEs) from both forecasting approaches and for all 100 forecasts are shown in Figure 4.4. Referring to the MSE results shown in Figure 4.4 and the statistics reported above the forecasting results for each of the four markets can be summarized as follows.

Stock Market

The MSEs for the stock market (upper panel in Figure 4.4) are in line with the forecasting statistics reported above. For the one- and two-period ahead forecasts, the GARCH model produces a lower median MSE. For horizons three to six, boosting produces lower median MSEs. However, for all horizons, the GARCH forecasts have a higher dispersion, as is reflected in the boxplots by their central 75%–quantiles. Thus, not only does the boosting approach provide better medium- and long-term volatility predictions for the S&P 500, its predictions are more robust for all horizons leading to less extreme MSEs than the GARCH benchmark.

Commodity Market

As the boxplots (second from top in Figure 4.4) show, for all horizons, out-of-sample boosting forecasts outperform, on average, those of the GARCH model. Theil's U (see above) supports this result. For all horizons, except the first, Theil's U is below 1. For six-month-ahead forecasts, Theil's U decreases to 0.714. This indicates that the boosting model strongly outperforms the GARCH model, especially for medium and long horizons. The modified Diebold and Mariano (1995b) test confirms this.

Bond Market

For the short-term prediction MSEs for the bond market (third panel in Figure 4.4) there is a neck and neck race, while, for the longer horizons, boosting tends to

[10] It should be noted that, when reverting the hypothesis (i.e., the null states boosting performs better than the benchmark), we obtain insignificant results in all cases.

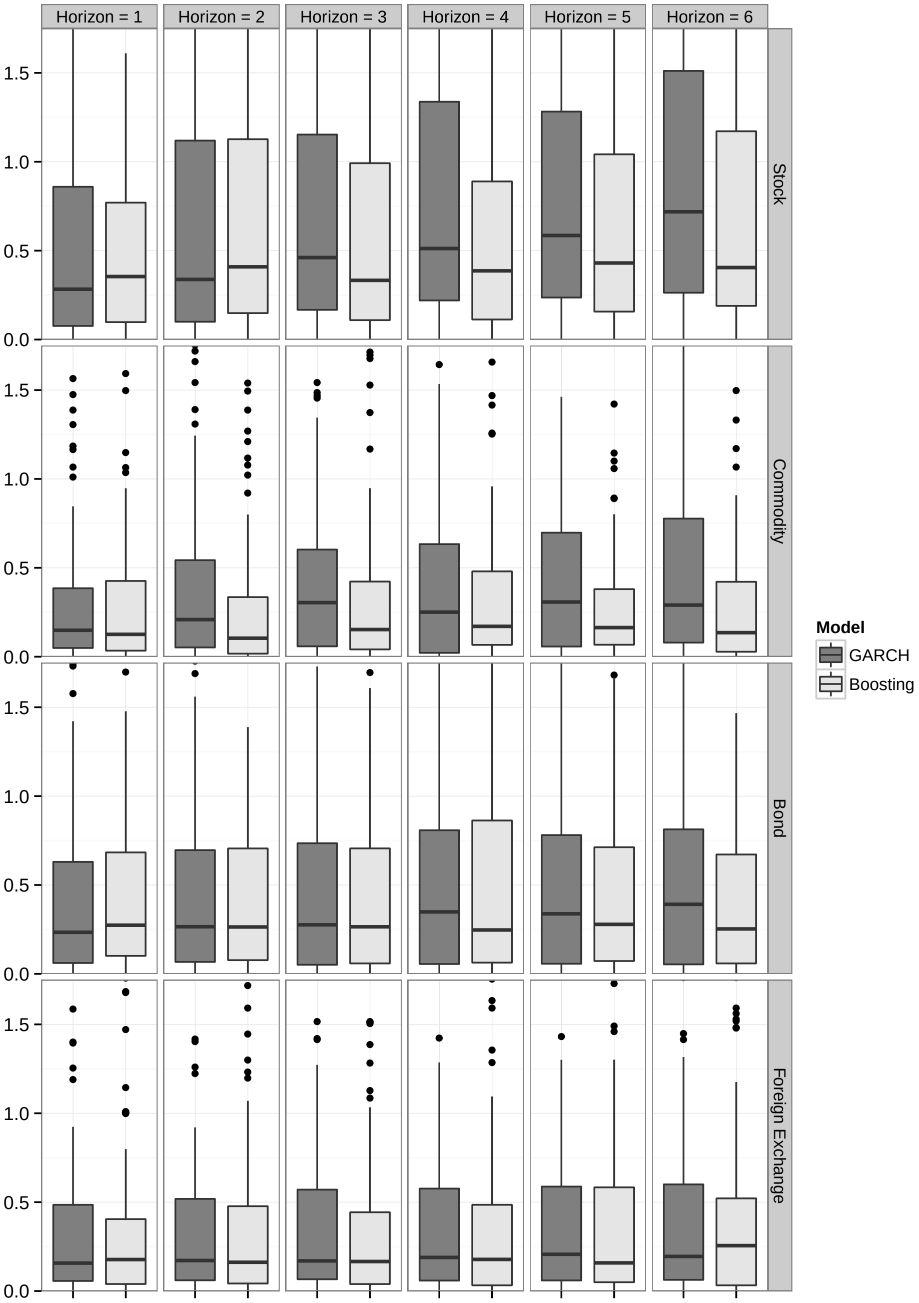


Figure 4.4: Comparison of forecasting MSEs between GARCH(1,1) and boosting for all markets.

deliver better forecasts on average. This also follows from Table 4.4: Theil's U is smaller than unity for horizons four, five, and six. However, according to the modified Diebold-Mariano test (Table 4.5), only in the six-month horizon boosting significantly outperforms GARCH.

Foreign Exchange Market

In line with the literature (e.g. Jorion, 1995; Nowak and Treepongkaruna, 2008), it appears to be difficult to provide a useful model to predict FX volatility. This is especially true for the lower frequency involved in using monthly observations. The low signal-to-noise ratio makes longer horizons in this market unpredictable. Still, boosting predictions are on the same level as those of the GARCH model forecasts (bottom panel in Figure 4.4). For horizons of three to six months, Theil's U is below 1, but none of the tests were significant. However, for all six horizons boosting leads to lower 75% MSE-quantiles, clearly suggesting a higher robustness of boosting forecasts.

4.4.2 The Driving Factors

From an economic viewpoint it is of interest to identify the financial and macroeconomic factors that drive financial market risk and to assess the specific manner in which these factors affect volatility.

A better knowledge about the driving forces for market volatility could be used for early warnings about market instabilities as well as for developing stabilization strategies. Therefore, the interpretation we gain is particularly advantageous when compared to the black-box nature of the GARCH framework. The following insights into the nature of volatility are based on a single one-period ahead model—as defined in Equation (4.1)—and are based on the whole data set. The driving forces will be summarized next, for each of the four markets.

Stock Market

Modeling the volatility of the S&P 500 with regression trees, we identify as the main drivers (lagged) IMS's U.S. financial stress index (FSI), the relative bond rate (RBR), lagged volatility, returns, the U.S. market excess return, and the CRB spot index. Figure 4.5 shows the impact of three relevant factors lagged once and twice.

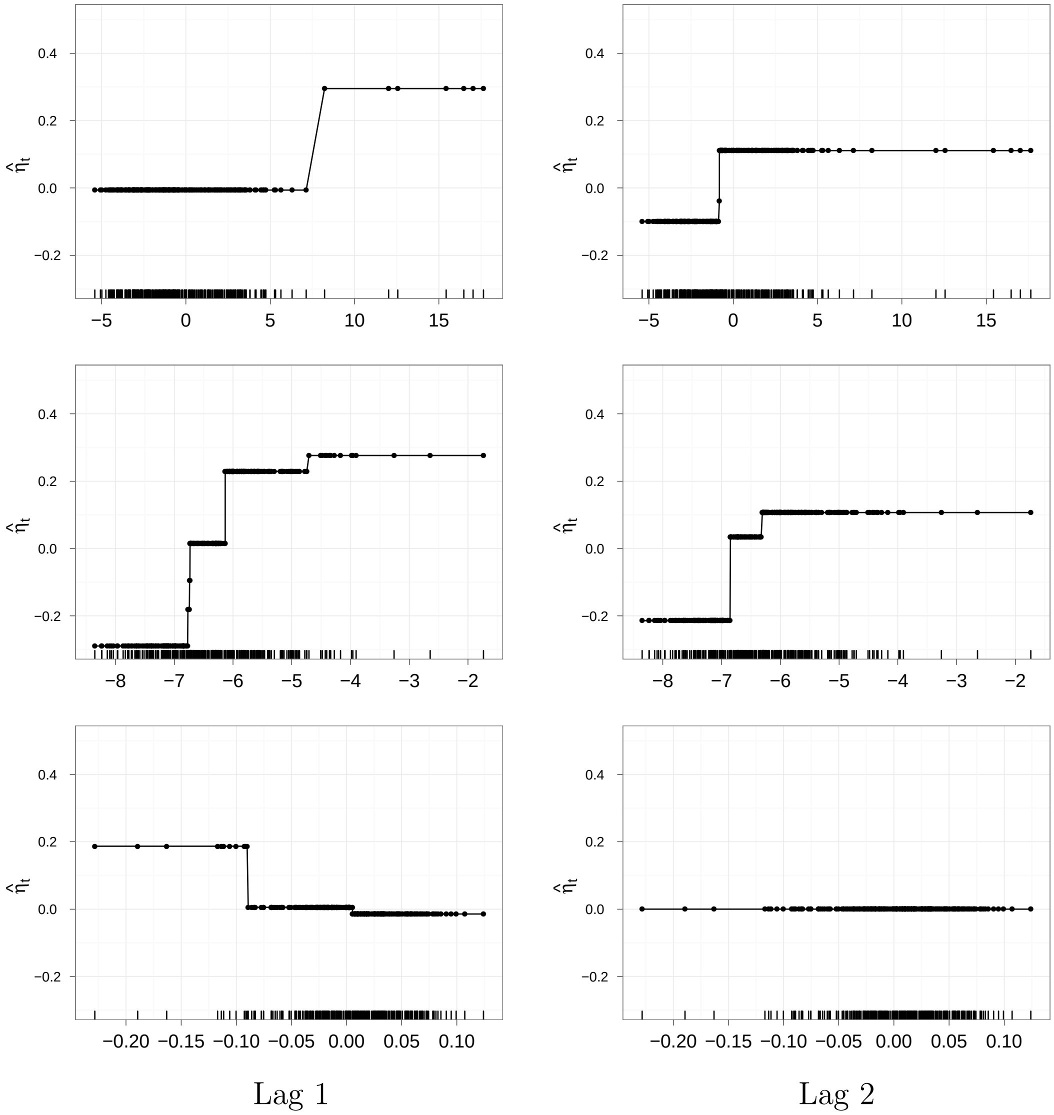


Figure 4.5: Three highly relevant volatility drivers for the S&P 500. Each row shows the coefficients for the first and the second lag of the FSI, RV, and S&P 500 returns, respectively.

The built-in variable (and lag) selection in our approach excluded all other potential drivers. Note that not all lags of the variables included are considered to be influential. For example, the lagged returns, the U.S. market excess return and the CRB spot index enter only through their first lag, whereas the FSI and the realized volatility have a greater, longer-lived impact, entering also with an additional second lag.

The IMF Financial Stress Index for the U.S. aggregates seven variables capturing market stress in three financial market segments, namely, banking, securities markets, and foreign exchange markets. Its motivation and composition are discussed in Cardarelli et al. (2009). Figure 4.5 (upper panel) clearly shows regime-dependence of for the FSI's impact on volatility. FSI–values above 7.5 increase next month's stock market volatility (in log scale) by about 0.3, which corresponds to an increase of about 16%. FSI-values below 7.5 do not affect next month's volatility. As for the second lag, our results indicate that positive (negative) FSI–values moderately increase (decrease) volatility in two months in a more or less symmetric fashion.

Another finding is that (log) realized volatility depends nonlinearly on past realized volatility. As shown (middle panel in Figure 4.5), small values of realized volatility, i.e., $\mathrm{RV} < -7$ or $\exp(\mathrm{RV}) < 0.03$, cause a decrease in next month's volatility. From approximately $\mathrm{RV} > -6$ onward, the influence becomes positive, i.e., the volatility is expected to increase in a highly nonlinear fashion. As Figure 4.5 suggests, the two-month impact is also nonlinear. Values of $\mathrm{RV} < -7$ will reduce volatility by about 10% and values of $\mathrm{RV} > -6.3$ result in an increase of about 5%.

Furthermore, we find that positive changes in the S&P 500 index slightly decrease volatility, whereas small negative changes (between −10% and 0%) moderately increase volatility (Figure 4.5, bottom panel). On the other hand, large negative returns (below −10%) increase volatility by about 10%. Finally, the relative bond rate (RBR) entails a considerable increase in volatility by about 28%, when it increases above one percent. Positive U.S. market excess returns have a moderate calming effect on the market, whereas values below −2.5% increase the volatility by 2%.

Commodity Market

The volatility of the commodity market is influenced by the past realized volatility, the net equity expansion, the Cochrane Piazessi factor, and the U.S. market excess returns. The Cochrane Piazessi factor impacts through both the first and the second lag, whereas the net equity expansion influences only through the second lag.

Figure 4.6 (upper panel) reveals that realized volatility depends in a highly nonlinear fashion on its first lag. Highly negative values of lagged realized volatility (below −6.5) dampen volatility by roughly 0.2 on the log scale which translates to a drop in volatility by about 10%. Values above −6.5 lead to an increase of the

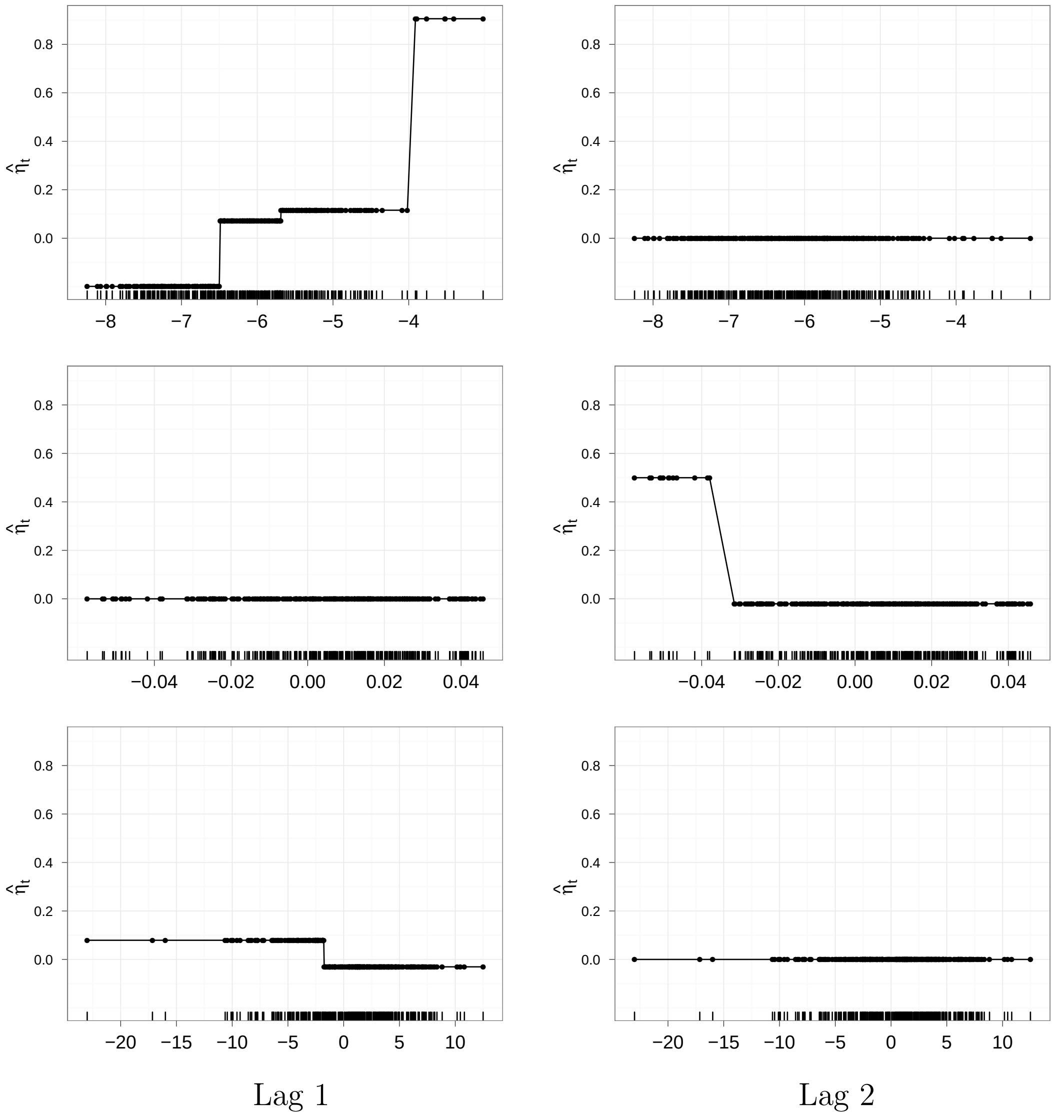


Figure 4.6: Three highly relevant volatility drivers in the commodity market. Each row shows the first and the second lag of the RV, net equity expansion, and the U.S. market excess return, respectively.

volatility in the commodity market. At −4 there is a jump, beyond which volatility increases by about 60%. Net equity expansion (Figure 4.6, center panel) has an increasing effect on volatility, if it is below −3%; otherwise it slightly decreases volatility. U.S. market excess returns above −2% dampen volatility, values below that increase volatility (Figure 4.6, bottom panel). The pattern is similar for the Cochrane Piazessi factor, except that the threshold there is at 2%.

Bond Market

When modeling the base functions with regression trees, we find that in the bond market volatility is driven by the default spread, the change of the money supply (M1), the changes in the purchasing manager index, net equity expansion, the relative bond rate, the change in consumer sentiment, and the book–to–market ratio.[11]

For the influence of the default spread (Figure 4.7, top panel), we find two clearly distinct regimes: a default spread above 1.1% tends to increase volatility by 7% in the following month, and values below that threshold reduce volatility by roughly 4%. The relative bond rate has an effect on volatility only if it exceeds 1%, in which case it increases bond volatility by 10%. A change in consumer sentiment or the book–to–market ratio produces a similar pattern: below a certain threshold—5% for consumer sentiment and 0.72 for the book–to–market ratio—they have no influence on volatility. Only if they exceed these thresholds they induce a rise in volatility. Sizable increases in M1 (above 5%) let volatility grow by approximately 10%. Smaller expansions or reductions in M1 decrease the volatility by 6.8% (Figure 4.7, center panel).

Foreign Exchange Market

A large number of factors seem to drive FX volatility. They include the FSI, the default spread, realized volatility, the TED spread, the U.S. market excess return, the long-term rate of return, and changes in M1. Periods of high financial stress, with the FSI assuming values above five, drive up volatility by 12%, whereas low financial stress reduces it, though, by a much smaller amount, namely less than 1% (Figure 4.8, top panel). Similar to the other markets, once-lagged realized volatility below -7 on the log scale, or $\hat{\sigma}^2 < 3\%$, lowers volatility marginally (Figure 4.8, center panel). Values above this cutoff boost volatility by 15%. U.S. market returns seem to influence volatility only if they are below -10%, in which case they increase the volatility.

[11] Bond return volatility has not been extensively studied in the literature. Two exceptions are Huang, Lu, and Wu (2011) and Viceira (2012).

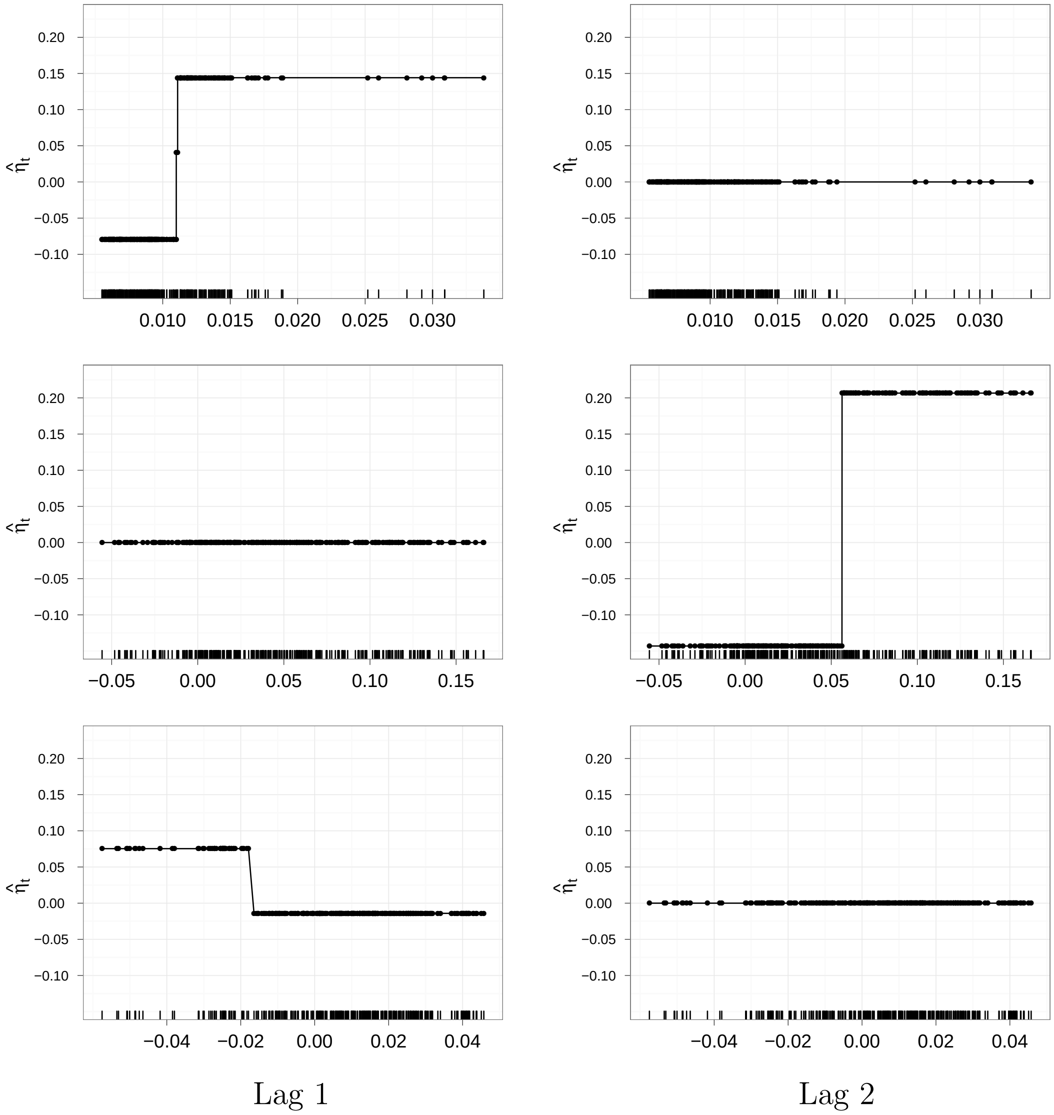


Figure 4.7: Three highly relevant volatility drivers in the bond market. Each row shows the first and the second lag of the default spread, change of M1, and the net equity expansion, respectively.

4.5 Conclusions

We have analyzed the determinants of volatility in four broad asset classes, namely, stocks, commodities, bonds, and foreign exchange, employing a wide range of poten-

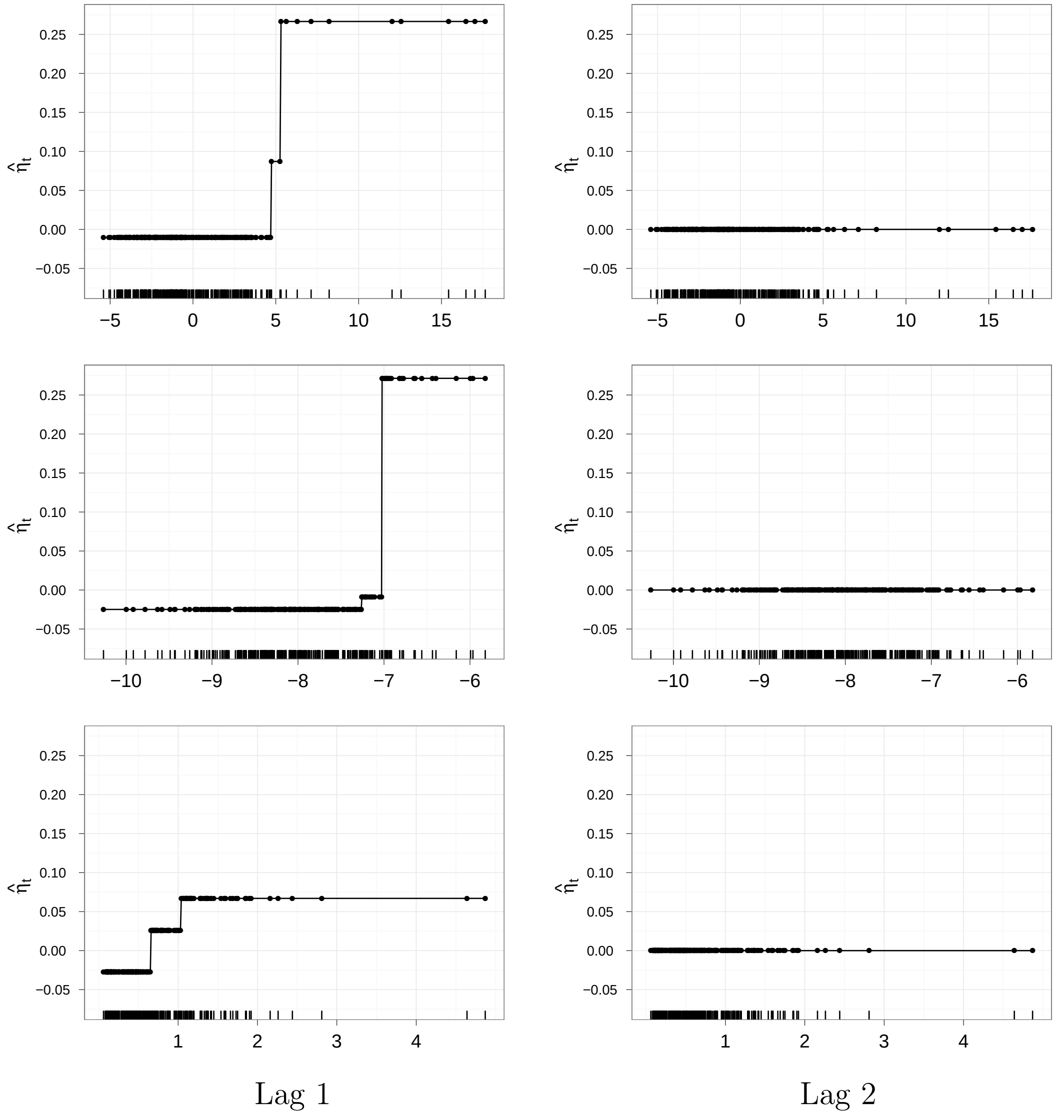


Figure 4.8: Three highly relevant volatility drivers in the Foreign Exchange market. Each row shows the first and the second lag of the FSI, RV, and the TED spread, respectively.

tial macro and financial drivers. Using monthly data, we adopted boosting techniques based on regression trees as base learners to identify relevant volatility drivers as well as the functional form of their influence. Specifically, we used componentwise boosting, which is tailor-made for sorting out irrelevant (lagged) predictors. First– and second–order lags of all drivers were included—along with some (seasonal) deterministic components—in a regression–type model.

Our empirical results give insight into the "anatomy" of volatility by identifying small groups of influential drivers for each market and by estimating driver–specific thresholds, which partition its domain into areas with similar impacts on volatility. By doing so, nonlinear dependencies can be identified. We do, indeed, find highly nonlinear influences of financial drivers on volatility. This contrasts the existing literature, which has almost exclusively concentrated on linear volatility dynamics.

Out-of-sample forecast using realized volatility as a proxy for the unobserved volatility suggests that the boosting approach performs very favorable for stocks and commodities relative to the common GARCH(1,1) benchmark model. The advantages are particularly convincing for longer forecasting horizons. For the bond and foreign exchange markets, boosting offers a similar short–term and a marginally better medium– to long–term accuracy. In all cases, however, boosting leads to more robust, i.e., less outlier–prone prediction errors than the GARCH benchmark.

Our findings suggest that boosting is well suited for a unified framework to predictor selection and estimation of volatility models in the presence of many potential (and possibly highly correlated) risk drivers. An advantage of the approach is that it can cope with "wide" data situations (Hastie et al., 2009a), i.e., situations in which the number of predictors exceed the number of observations.[12] Models obtained in this fashion can be a starting point for more detailed nonlinear model specifications, but could also be used in certain financial applications, such as dynamic portfolio optimization or option valuation.

[12] In the application presented here, we typically had 58 predictors and 230 observations.

Chapter 5

Boosting for Estimating Spatially Structured Additive Models

Spatially structured additive models offer the flexibility to estimate regression relationships for spatially and temporally correlated data. Here, we focus on the estimation of conditional deer browsing probabilities in the National Park "Bayerischer Wald." The models are fitted using a componentwise boosting algorithm. Smooth and nonsmooth base learners for the spatial component of the models are compared. A benchmark comparison indicates that browsing intensities may be best described by nonsmooth base learners, allowing for abrupt changes in the regression relationship. This chapter is based on Robinzonov and Hothorn (2010).

5.1 Introduction

Biological diversity and forest health are major contributors to the ecological and economic prosperity of a country. This is what makes the conversion of mono-species into mixed-species forests an important concern of forest management and policy in Central Europe (Knoke, Ammer, Stimm, and Mosandl, 2008). Recent research shows that there are not only positive ecological effects of mixed-species forests (e.g. Fritz, 2006) but also positive economic consequences (Knoke and Seifert, 2008). Like any other living environment, the development of forests is strongly conditioned on a balanced and consistent regeneration. Whether natural or artificial, the regeneration is hindered at a very early stage by browsing damage caused by various game species. In middle Europe, especially, roe and red deer are the most common species

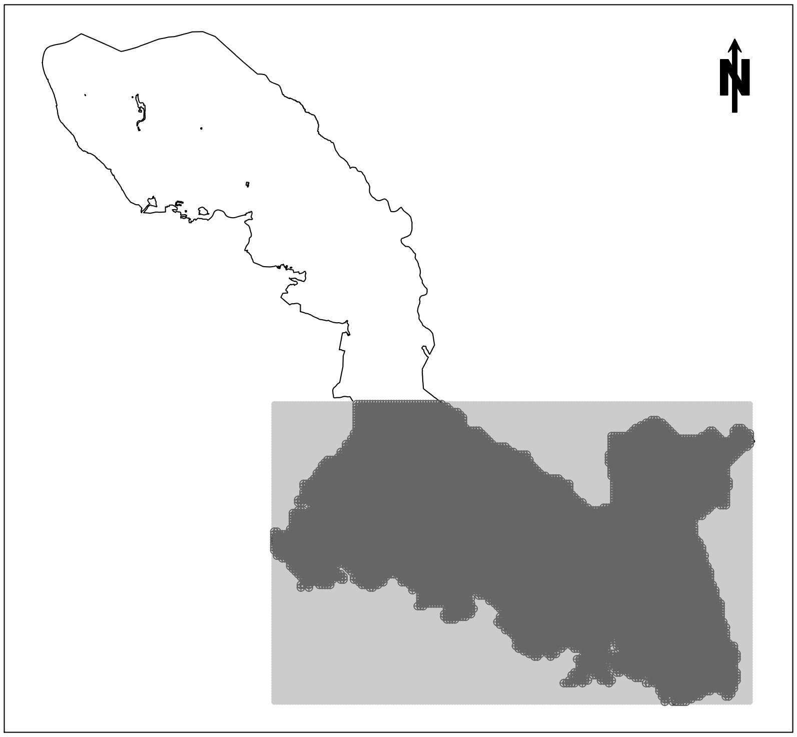


Figure 5.1: The National Park "Bayerischer Wald." The southern gray colored region is the district of "Rachel-Lusen" where our studies take place.

browsing on young trees. This activity is certainly natural by definition. However, the eradication of large predators, the conversion of the landscape, and the fostering of trophy animals have given rise to increased numbers of deer and so to intensified browsing pressure over the past centuries. The consequences of excessive browsing often lead to retardation and homogenization of forest growth (Eiberle and Nigg, 1987; Eiberle, 1989; Ammer, 1996; Motta, 2003; Weisberg, Bonavia, and Bugmann, 2005).

Forest regeneration is monitored on a regular basis by the Bavarian Forest Administration (Forstliches Gutachten, 2006). This Bavarian-wide survey is conducted every three years and takes place in all 745 game management districts (Hegegemeinschaften) in Bavaria. Preventive measures are proposed following the survey's results. In case of an estimated browsing quota above the specified thresholds, the local authorities consider how to protect the most vulnerable areas. An often used practice is to recommend intensified deer harvesting in the corresponding areas. Whether the impact of game on the forest regeneration is correctly measured remains a matter of debate (e.g. Prien, 1997; Rüegg, 1999; Moog, 2008). Developing precise measures which reflect the true condition of the forest's regeneration is thus crucial and nontrivial.

Our focus is on surveys conducted to estimate the local conditional probability of a young tree to be affected by deer browsing, as recommended for monitoring of the influence of game on forest regeneration (Rüegg, 1999). For the beech species (*Fagus sylvatica*) this quantity reflects the exposure to deer browsing and is the basis for subsequent management decisions. Here, we are concerned with the estimation of such conditional browsing probabilities. We evaluate and compare boosting algorithms for fitting structured additive models (Fahrmeir et al., 2004) to deer browsing intensities. This research aims to make, in a brief presentation, a comparison of smooth and nonsmooth model components for capturing the spatio-temporal variation in such data. Our investigations are based on two surveys conducted in 1991 and 2002 in the district of "Rachel-Lusen," the southern part of the National Park "Bayerischer Wald" depicted in Figure 5.1.

5.2 Methods

The main purpose of a deer browsing survey is to estimate the probability of deer browsing on young trees. More specifically, the conditional probability of a young tree of a certain species at a given location to suffer from deer browsing is the quantity of interest. The tree height is an important exploratory variable for deer browsing and thus needs to be included in the model. In addition, unobserved heterogeneity in the browsing damage will be considered by allowing for spatial and spatio-temporal components to enter the model. Commonly, other covariates describing the forest ecosystem are not measured and are thus not included in our investigations. For the sake of simplicity, we restrict our attention to beeches. Below we will sketch our model in (5.1), giving a general view of the estimation strategy. In the subsequent sections we will consider the component pieces of three possible approaches meant to accomplish this strategy.

The general idea of our modeling strategy is as follows. The logit transformed conditional probability of browsing damage is linked to the tree height, the spatial, and spatio-temporal, effects by the regression function f such that

$$\begin{aligned}\operatorname{logit}(\mathbb{P}(Y=1|\mathsf{height},\ \mathsf{space},\ \mathsf{time})) &= f(\mathsf{height},\ \mathsf{space},\ \mathsf{time}) \\ &= f_{\text{height}}(\mathsf{height}) + f_{\text{spatial}}(\mathsf{space}) \\ &+ f_{\text{spatemp}}(\mathsf{space}, \mathsf{time}),\end{aligned} \tag{5.1}$$

where the predictor space represents a two-dimensional covariate of $\mathsf{northing}$ and $\mathsf{easting}$, height is a one-dimensional continuous variable representing tree height, and time is an ordered factor with levels $\mathsf{1991}$ and $\mathsf{2002}$.

Therefore, we differentiate between three types of variability: that caused by tree height and captured by f_{height}, solely spatial variability explained by the two-dimensional smooth function f_{spatial}, and time-dependent heterogeneity modeled by the multi-dimensional smooth function f_{spatemp}.

5.2.1 Spatio-Temporal Structured Additive Models

The estimation of the first two models is carried out by boosting. Kneib et al. (2009) introduce smooth P-spline tensor products in the context of boosting. We apply this idea below and compare it to a new proposition for spatial estimation based on regression trees.

These two boosting methods differ solely in the choice of their spatial and spatio-temporal base procedures. The first boosting method is a structured additive regression (GAMBoost) model for describing the probability of browsing damage:

$$\begin{aligned}\operatorname{logit}(\mathbb{P}(Y=1|\mathsf{height},\ \mathsf{space},\ \mathsf{time})) &= f_{\text{str}}(\mathsf{height},\ \mathsf{space},\ \mathsf{time}) \\ &= f_{\text{bheight}}(\mathsf{height}) + f_{\text{bspatial}}(\mathsf{space}) \\ &+ f_{\text{bspatemp}}(\mathsf{space}, \mathsf{time})\end{aligned} \tag{5.2}$$

where f_{bheight} is an additively structured, P-spline function of height, f_{bspatial} is an additively structured, bivariate P-spline tensor function of $\mathsf{easting}$ and $\mathsf{northing}$ (or, for short, space), and f_{bspatemp} is essentially the same as f_{bspatial} but applied only for the year 2002 (see (5.7) below). The objective is to obtain an estimate $\hat{f}_{\text{str}}$ of the function f_{str}. In theory, this approximation is usually based on the expectation of some prespecified loss function $L(y, \pi(f_{\text{str}}))$; in practice we aim at minimizing its empirical version

$$\hat{f}_{\text{str}} = \arg\min_{f_{\text{str}}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, \pi_i(f_{\text{str}})) \tag{5.3}$$

where $\pi_i(f_{\text{str}}) = \text{logit}^{-1}(f_{\text{str}}(\mathsf{height}_i, \mathsf{space}_i, \mathsf{time}_i))$ denotes the inverse of the logit function. The object is to minimize the negative log-likelihood

$$L(y_i, \pi_i(f_{\text{str}})) = -(y_i \ \log(\pi_i(f_{\text{str}})) + (1 - y_i) \ \log(1 - \pi_i(f_{\text{str}}))). \tag{5.4}$$

As mentioned above, each function in (5.2) has an additive structure. This means, in particular, that the model can be decomposed into

$$\hat{f}_{\text{bheight}}(\mathsf{height}) = \nu \sum_{m=0}^{M} \hat{h}^{[m]}_{\text{height}}(\mathsf{height}) \tag{5.5}$$

$$\hat{f}_{\text{bspatial}}(\mathsf{space}) = \nu \sum_{m=0}^{M} \hat{h}^{[m]}_{\text{spatial}}(\mathsf{space}) \tag{5.6}$$

$$\begin{aligned} \hat{f}_{\text{bspatemp}}(\mathsf{space}, \mathsf{time}) &= \nu \sum_{m=0}^{M} \hat{h}^{[m]}_{\text{spatemp}}(\mathsf{space}, \mathsf{time}) \\ &= \begin{cases} \nu \sum\limits_{m=0}^{M} \hat{h}^{[m]}_{\text{spatial}}(\mathsf{space}), & \mathsf{time} = 2002, \\ 0, & \mathsf{time} = 1991, \end{cases} \end{aligned} \tag{5.7}$$

where the base learner $\hat{h}^{[m]}_{\text{height}}$ is a smooth penalized B-spline function (P-spline, Eilers and Marx, 1996), $\hat{h}^{[m]}_{\text{spatial}}$ is a smooth bivariate P-spline based surface and $\nu \in (0, 1)$ is the shrinkage parameter. Thus, our choice of base learners are basis expansions of the form

$$\hat{h}^{[m]}_{\text{height}}(\mathsf{height}) = \sum_{k=1}^{K} \hat{\gamma}^{[m]}_{\text{height},k} \, b_k(\mathsf{height}) \tag{5.8}$$

$$\hat{h}^{[m]}_{\text{spatial}}(\mathsf{space}) = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \hat{\gamma}^{[m]}_{\text{space}_{k_1,k_2}} \, b_{k_1,k_2}(\mathsf{space}) \tag{5.9}$$

where the b_k's represent K completely known univariate basis functions, b_{k_1,k_2} tensor product functions with

$$b_{k_1,k_2}(\mathsf{space}) = b_{k_1,k_2}(\mathsf{easting}, \mathsf{northing}) = b_{k_1}(\mathsf{easting}) b_{k_2}(\mathsf{northing})$$

and $\hat{\boldsymbol{\gamma}}^{[m]}_{\text{height}}$ and $\hat{\boldsymbol{\gamma}}^{[m]}_{\text{space}}$ are regression coefficients which scale these basis functions (see Wood, 2006; Kneib et al., 2009). $\hat{\boldsymbol{\gamma}}^{[0]}_{\text{height}}$ and $\hat{\boldsymbol{\gamma}}^{[0]}_{\text{space}}$ are arbitrarily chosen start vectors of parameters. Note that the time-dependent effect in (5.7) is interpreted as the spatial difference between the years 1991 and 2002. It should be further noted that K, K_1, and K_2 are known in advance (specified by the user), and M is the major tuning parameter for boosting which we discuss below.

All parameters, i.e., all $\hat{\boldsymbol{\gamma}}^{[m]}$s, of this additive expansion will be determined iteratively by successively improving (updating) them and accumulating the whole estimation in $\hat{f}_{\text{str}}$. Hence, the step size ν can be thought of as an improvement penalty which prevents the model from taking the full contribution of the updates.

The minimization problem (5.3) is solved iteratively by componentwise boosting which chooses at each step the "best" base procedure from (5.5)–(5.7), i.e., the one that most contributes to the fit. One option to attain this is via the steepest-descent optimization, which relies on the negative gradient

$$g_i = -\left[\frac{\partial}{\partial f_{\text{str}}} L(y_i, \pi_i(f_{\text{str}}))\right]_{f_{\text{str}}=\hat{f}_{\text{str}}}, i = 1, \ldots, n \tag{5.10}$$

being computed at each step and subsequently fitted against each base procedure separately, i.e., the negative gradient is used as a *pseudo-response* in each step m. The negative gradient (5.10) indicates the direction of the locally greatest decrease in the loss. The most "valuable" covariate has the highest correlation with the negative gradient and is therefore chosen for fitting.

Schmid and Hothorn (2008) carried out an extensive analysis of the effects of the main hyper-parameters on boosting, such as the maximum step number M, the step size ν, the smoothing parameters for the P-splines, and the number of knots. Their results confirmed the common knowledge that there exist a minimum number of necessary knots needed to capture the curvature of the function and that the algorithm is not sensitive to this choice (20–50 knots should be sufficient). They also found that $\nu = 0.1$ is a reasonable choice for the step size, altering which interacts only with the computational time, i.e., smaller ν increases the computational burden but does not deteriorate the fitting quality. The same holds for the P-spline smoothing parameters, which essentially penalize the flexibility of the base procedure through its degrees of freedom. Choosing larger values leads to fewer degrees of freedom, which translates into larger bias but smaller variance. This follows the prescriptions of the recommended strategy for boosting (Bühlmann and Yu, 2003; Schmid and Hothorn, 2008). Again, reasonable alteration of this parameter solely impacts the computational time.

Aside from obtaining the stopping condition M (which will be discussed later), we are ready to summarize componentwise boosting in the following algorithm:

```
Componentwise boosting

1. Initialize $\hat{f}_{\text{str}}$ = offset, set $m = 0$.
2. $m = m + 1$.
3. Compute the negative gradient: $g_i = -\left[\frac{\partial}{\partial f_{\text{str}}} L(y_i, \pi_i(f_{\text{str}}))\right]_{f_{\text{str}}=\hat{f}_{\text{str}}}, i = 1, \dots, n$.
4. Fit all base procedures to the negative gradient and select the best one according to
   $\hat{s}_m = \underset{s\in\{\text{height, spatial, spatemp}\}}{\arg\min} = \sum_{i=1}^{n}(g_i - \hat{h}_s^{[m]})^2$.
5. Update $\hat{f}^{\text{str}} := \hat{f}^{\text{str}} + \nu\,\hat{h}_{\hat{s}_m}^{[m]}$.
6. Iterate 2–5 until $m = M$.
```

Researchers in many fields have found the cross-validatory assessment of tuning parameters attractive. By splitting the original (training) set into k roughly equally sized parts, one can use $k-1$ parts to train the model and the last, kth, part to test it. This is known as a k-fold cross-validation. A known issue of cross-validation is the systematic partition of the training set increasing the risk of error patterns. That is, the training set is not a random sample from the available data, but chosen to disproportionally represent the classes, especially to over-represent rare classes. Therefore, we alleviate this to some degree by using the bootstrap algorithm (Efron, 1979). We perform a random sampling with replacement of the original data set, i.e., the n sample points are assumed to be multinomially distributed with equal probability $1/n$. After the sampling, we have a new training set of size n with some sample points chosen once, some more than once, and some of them being completely omitted (usually $\sim 37\%$). Those omitted sample points are regarded as our test set in order to quantify the performance. We choose some large value for M, say 2000, and perform 25 bootstrapped samples with each $m = 1, \dots, M$. The optimal m is reported according to the average out-of-sample risk, also referred to as *out-of-bag*, minimization of the loss function.

5.2.2 Tree Based Learners

There are regions in the National Park “Bayerischer Wald” which are not affected by deer browsing and others with disproportionally higher risks of browsing. This is due to the irregular distribution of regeneration areas in the National Park and to other environmental factors, e.g., populated regions. Therefore, we might wish to reconsider the smooth relationship between the response and the predictors made so far. We aim to improve the performance of regression setting (5.2) by reconsidering the smooth assumption of the underlying function f_{str}. Having covariates at different scales, we find regression trees (Breiman et al., 1984) to be an attractive way to express knowledge and aid forest decision-making. A "natural" candidate for a decision tree based learner is the spatio-temporal component due to the different scales of space and time. The spatial component space is another good option for a tree based modeling due to the coarse relationship between the space and the browsing probability which we suspect. We let the smooth P-spline based learner h_{height} remain unchanged. Therefore, we have a similar structure to (5.2)

$$\begin{aligned}\text{logit}(\mathbb{P}(Y=1|\mathsf{height},\ \mathsf{space},\ \mathsf{time})) &= f_{\text{bb}}(\mathsf{height},\ \mathsf{space},\ \mathsf{time}) \\ &= f_{\text{bheight}}(\mathsf{height}) + f_{\text{bbspatial}}(\mathsf{space}) \\ &+ f_{\text{bbspatemp}}(\mathsf{space}, \mathsf{time})\end{aligned} \tag{5.11}$$

with f_{bheight} being exactly the same as in (5.5) and modified learners

$$\hat{f}_{\text{bbspatial}}(\mathsf{space}) = \nu \sum_{i=1}^{M} \hat{h}^{[m]}_{\text{spatialtree}}(\mathsf{space}), \tag{5.12}$$

$$\hat{f}_{\text{bbspatemp}}(\mathsf{space}, \mathsf{time}) = \nu \sum_{i=1}^{M} \hat{h}^{[m]}_{spatemptree}(\mathsf{space}, \mathsf{time}). \tag{5.13}$$

The model (5.11) is referred to as the TreeBoost model. We choose the unbiased recursive partitioning framework of Hothorn et al. (2006) to grow binary trees. The spatial component has the additive form

$$\hat{h}^{[m]}_{\text{spatialtree}}(\mathsf{space}) = \sum_{j=1}^{J} \hat{\gamma}^{[m]}_{\text{spatialtree},j}\ I(\mathsf{space} \in R^{[m]}_j) \tag{5.14}$$

and the spatio-temporal component is represented by

$$\hat{h}^{[m]}_{\text{spatemptree}}(\mathsf{space}, \mathsf{time}) = \sum_{j=1}^{J^*} \hat{\gamma}^{[m]}_{\text{spatemptree},j}\ I((\mathsf{space}, \mathsf{time}) \in R^{*[m]}_j), \tag{5.15}$$

where I denotes the indicator function, $R^{[m]}_j, j = 1, \ldots, J$ are disjoint regions which collectively cover the space of all joint values of the predictor variables in space (recall that $\mathsf{space} = \{\mathsf{easting}, \mathsf{northing}\}$). The superscript $[m]$ in $R^{[m]}_j$ means that this region is defined by the terminal nodes of the tree at the mth boosting iteration. $R_j^{*[m]}$ are the respective regions for space and time. Thus, we compute a sequence of simple binary trees with a maximal depth of, say, five. The task at each step is to find a new tree to describe the prediction residuals (the gradient) of the preceding tree succinctly. The next tree will then be fitted to the new residuals and will further partition the residual variance for the data, given the preceding sequence of trees.

5.2.3 Generalized Additive Model

The last method in our comparison is the generalized additive model (GAM) proposed by Hastie and Tibshirani (1990). Once we are familiar with the underlying structure of the GAMBoost model, the GAM model can be seen as a simplified special case of (5.2) with $\nu = 1$, $M = 1$, and with no componentwise selection carried out. This means that we have the following structure

$$\begin{aligned}\operatorname{logit}(\mathbb{P}(Y=1|\mathsf{height},\, \mathsf{space},\, \mathsf{time})) &= f(\mathsf{height},\, \mathsf{space},\, \mathsf{time}) \\ &= f_{\text{height}}(\mathsf{height}) + f_{\text{spatial}}(\mathsf{space}) \\ &+ f_{\text{spatemp}}(\mathsf{space}, \mathsf{time})\end{aligned} \tag{5.16}$$

where

$$\hat{f}_{\text{height}}(\mathsf{height}) = \sum_{k=1}^{K} \hat{\gamma}_{height,k}\, b_k(\mathsf{height}) \tag{5.17}$$

$$\hat{f}_{\text{spatial}}(\mathsf{space}) = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \hat{\gamma}_{\text{space}_{k_1,k_2}}\, b_{k_1,k_2}(\mathsf{space}) \tag{5.18}$$

$$\hat{f}_{\text{spatemp}}(\mathsf{space{,}time}) = \begin{cases} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \hat{\gamma}_{\text{space}_{k_1,k_2}}\, b_{k_1,k_2}(\mathsf{space}), & \mathsf{time} = 2002, \\ 0, & \mathsf{time} = 1991. \end{cases} \tag{5.19}$$

The interpretation of the basis function b_k, b_{k_1,k_2} and their scaling parameters remains the same as in (5.8) and (5.9). A similar model to (5.16) has been proposed by Augustin, Musio, von Wilpert, Kublin, Wood, and Schumacher (2009). A major difference between their model and the specification above is the time component being a continuous predictor smoothly modeled through cubic regression spline basis functions. This is what they call a 3-d tensor product smoother for space and time. It is also worth mentioning that their model prescinds from the pure spatial component f_{spatial} and relies solely on the multi-dimensional function f_{spatemp} to capture the spatial variability.

5.3 Results

In this section we apply the three models to spatial data collected from the National Park "Bayerischer Wald." We depict the surface of the estimated browsing probability and indicate the absolute number of browsing cases. In addition, we carry out a model comparison by means of an out-of-sample prediction of the likelihood function.

5.3.1 Spatial Estimates

In a first step, we visualize the browsing probability estimates obtained by the GAMBoost model (5.2), the TreeBoost model (5.11) and the common GAM as in (5.16). Figure 5.2 illustrates the estimation produced by the GAMBoost model for an average beech tree 60 cm in height. The light areas indicate regions with higher risk of browsing damage, the dark regions show areas with lower browsing probability. Furthermore, we use black circles proportional to the absolute number of damaged trees found in the corresponding location of the map.

The GAMBoost model proposes the smoothest fit of all the models. This model detects the risky regions in 2002 rather well, encompassing the black circles with smooth light regions and fitting the northern high-level areas to low risk probabilities. However, the relatively even empirical distribution of damaged cases in 1991 leaves the impression of too smooth a surface, i.e., possible underfitting. The GAMBoost model is also an example of why fine tuning of the hyper parameters should be undertaken with greater care in the presence of tensor P-spline base learners. The claims we made about the informal impact of the step size, the number of knots, and the smoothing parameters still hold in this case. However, the maximum number of boosting steps markedly increases if bivariate base learners are considered. One could falsely choose too small an M for an upper bound of the boosting steps. Therefore, boosting would continuously improve its prediction power within the proposed values of M and will always find the optimal M at the border, i.e., at the last step. This is due to the insufficient degrees of freedom leading to a very modest amount of progress towards optimality, i.e., the optimal step number is basically never reached. Therefore, the "standard" amount for degrees of freedom $\mathrm{df} \in (4, 6)$ for the univariate P-spline learners is insufficient for tensor P-spline learners. We use $\mathrm{df} = 12$ in order to speed up the computations and to ensure that $M = 2000$ is sufficient to find an optimal number of boosting iterations. Alternatively one could dampen the learning rate less severely by increasing the step size ν or altering the number of spline knots.

Figure 5.3 represents the estimation produced by the TreeBoost model for an

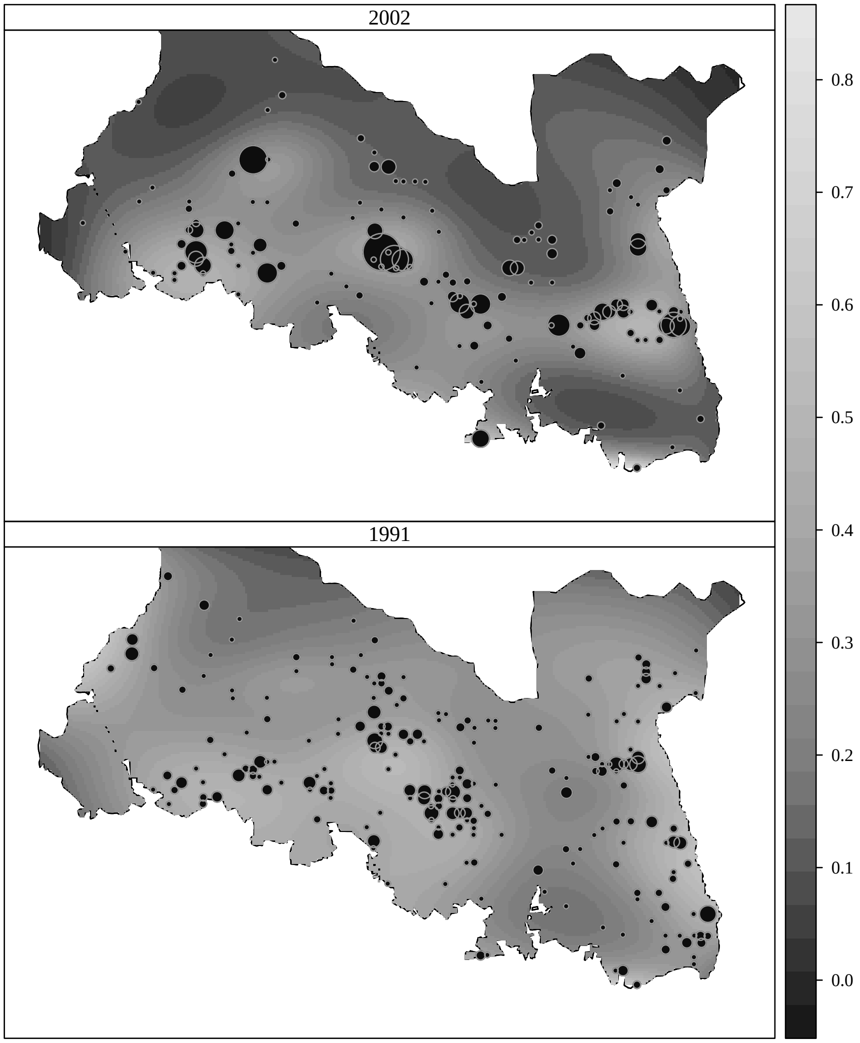


Figure 5.2: Spatial component space fitted by the GAMBoost model for an average beech tree at the height of 60 cm in the years 1991 (bottom) and 2002 (top). The diameter of the black circles is proportional to the absolute number of browsed trees.

average beech tree 60 cm in height. The color codes are the same as in the example above. The inherent coarse structure in the fit might look less attractive than Figure 5.2 but in the next section we will perform a formal bootstrap based model inference and will compare the predictive power of all the models in fair conditions.

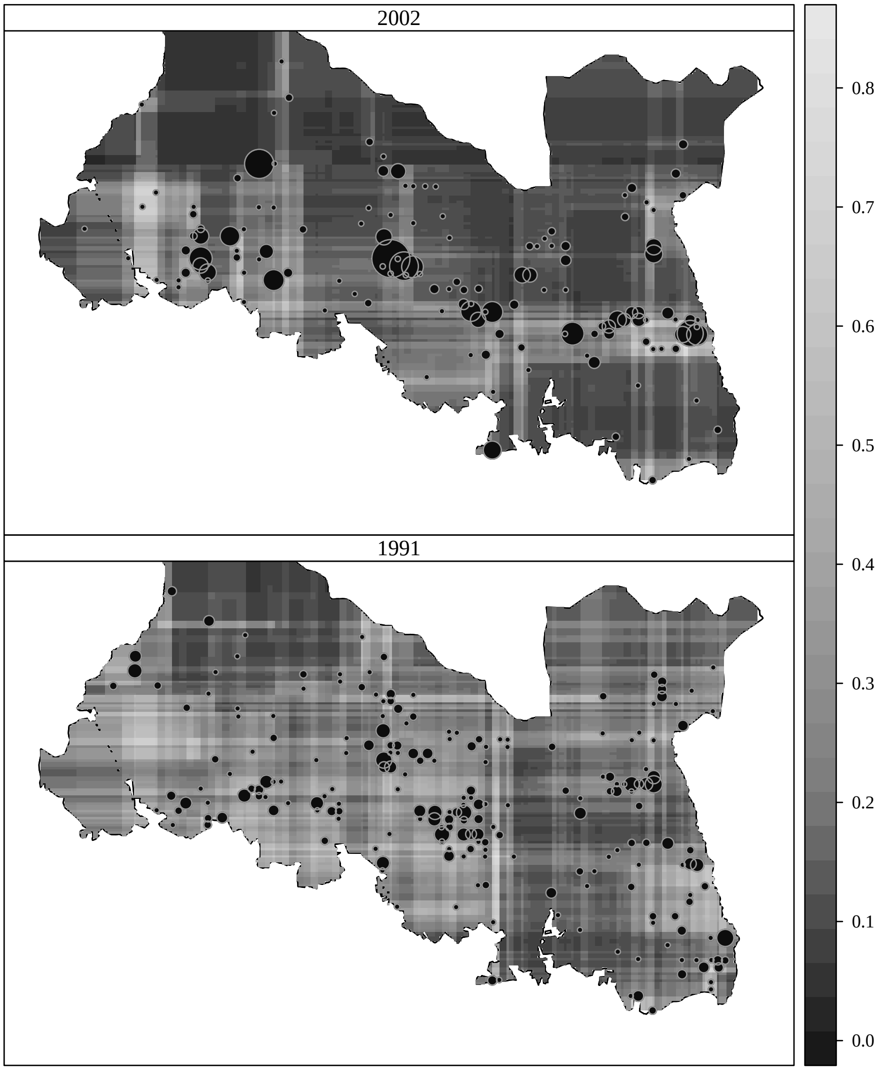


Figure 5.3: Spatial component `space` fitted by the TreeBoost model for an average beech tree 60 cm in height in the years 1991 (bottom) and 2002 (top). The diameter of the black circles is proportional to the absolute number of browsed trees.

Although not as straightforward as in Figure 5.2, the general pattern for the risky regions in the central and south-western parts of the National Park in 2002 remains visible. The final example is depicted in Figure 5.4 representing the GAM model. It

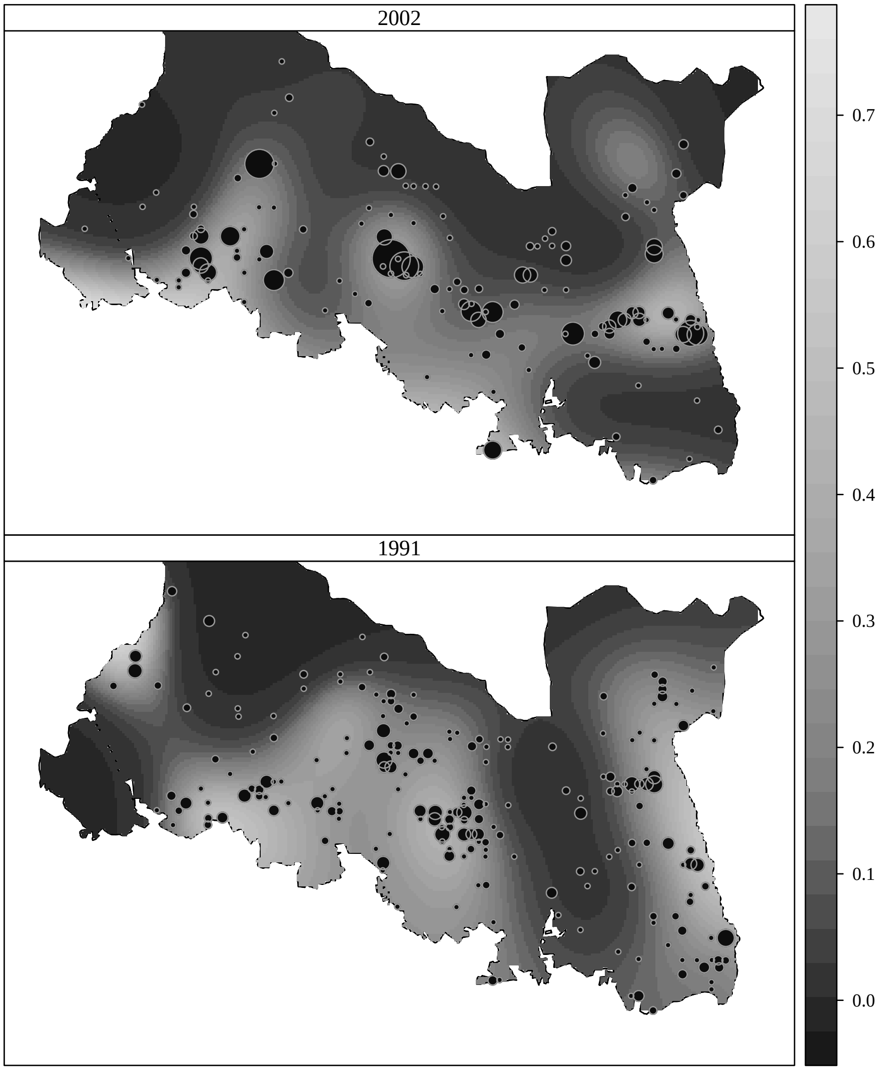


Figure 5.4: Spatial component space fitted by the Generalized Additive Model for a average 60 cm tree in the years 1991 (bottom) and 2002 (top). The diameter of the black circles is proportional to the absolute number of browsed trees.

proposes a similar structure to Figure 5.2 with nicely shaped smooth peaks in the risky areas. In the next section we carry out a model comparison of the predictive power of the different models.

5.3.2 Model Comparison

Eight models were fitted to the beech browsing data. Our three candidate models, GAMBoost (5.2), TreeBoost (5.11), and GAM (5.16) and their simplified versions, including several restrictions, are summarized in Table 5.1. The single column which requires additional clarification is the second column, termed "Label." The Label concisely represents the restrictions which we apply to the models. For instance, the label "A" refers to the simplest and fully constrained model with a single intercept as a covariate. "B" denotes a model which considers the height variable only, hence ignoring the spatial and the spatio-temporal effects. "C" means a model with the height predictor being constrained to zero and "D" denotes the most complex model, which considers all predictors.

We quantify the predictive power of each model using the out-of-bootstrap empirical distribution of the negative log-likelihood. For the boosting algorithms we, therefore, bootstrap twice: to find an optimal step number and to validate. For this, we divide the data set into a training and a validation part. In the first place, we bootstrap in the training sample, subdividing it into training and validation subsets. Therefore, we estimate the step number without touching the validation set. Secondly, we evaluate the negative log-likelihood with the estimated step number.

The results of the performance assessment are shown in Figure 5.5. Each boxplot represents 25 out-of-bootstrap values of the negative log-likelihood function based on the different models from Table 5.1. The first four light gray colored boxes represent the common GAM models. The highly constrained models "A" and "B" are not boosted and are primarily used to strengthen the credibility of the other models. The distinct risk collapse in all "C" models compared to "A" and "B" suggests the significant importance of the spatio-temporal effects on the browsing probability. It is further apparent that the height does contribute to the fit in the smooth specifications, i.e., "C" has a clearly higher risk than the largest model specification "D" for GAM and GAMBoost.

Further, we evidenced that boosting the smooth relationship between the response and the covariates is superior to the common GAM. This can be seen from the juxtaposition of the third and fourth light gray boxplots from the left and the two rightmost boxplots in Figure 5.5. Since all predictors are selected, the better prediction accuracy is solely explained by the effect of the shrinkage parameter.

It is instantly apparent that the TreeBoost model performs best compared to the other strategies. Thereupon, we empirically show that the assumption of a smooth underlying structure does indeed degrade performance in this case.

Table 5.1: An overview of all models under test.

Class	Label	Model Specification	Details
GAM	A	$f = 1$	An intercept model averaging the logits in the whole area of investigation.
	B	$f = f_{\text{height}}$	A model with restrictions $f_{\text{spatial}} = f_{\text{spatemp}} \equiv 0$ which allows only for the `height` effect.
	C	$f = f_{\text{spatial}} + f_{\text{spatemp}}$	A model with a restriction $f_{\text{height}} \equiv 0$ thus only allowing for spatial and spatio-temporal effects.
	D	$f = f_{\text{spatial}} + f_{\text{spatemp}} + f_{\text{height}}$	The full model defined in (5.16)
TreeBoost	C	$f_{bb} = f_{\text{bbspatial}} + f_{\text{bbspatemp}}$	A model with restrictions $f_{\text{bheight}} \equiv 0$ thus only allowing for spatial and spatio-temporal effects.
	D	$f_{bb} = f_{\text{bbspatial}} + f_{\text{bbspatemp}} + f_{\text{bheight}}$	The full model defined in (5.11).
GAMBoost	C	$f_{\text{str}} = f_{\text{bspatial}} + f_{\text{bspatemp}}$	A model with restrictions $f_{\text{bheight}} \equiv 0$ thus only allowing for spatial and spatio-temporal effects.
	D	$f_{\text{str}} = f_{\text{bspatial}} + f_{\text{bspatemp}} + f_{\text{bheight}}$	The full model defined in (5.2).

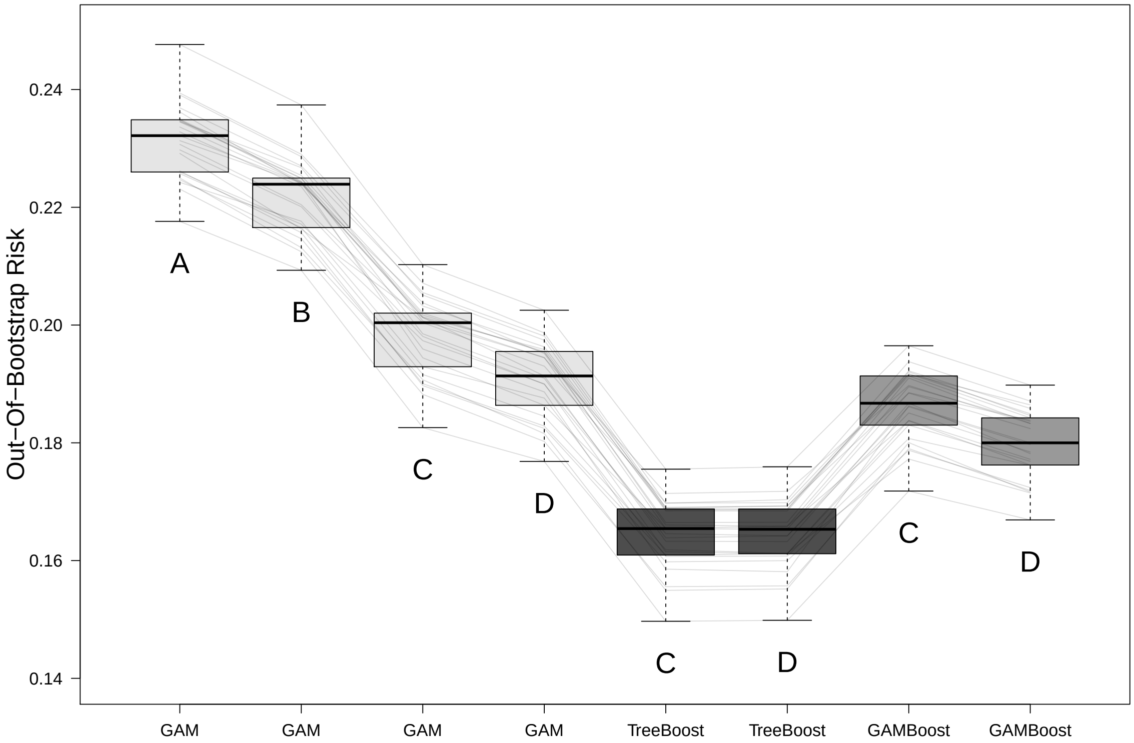


Figure 5.5: Out-Of-Bootstrap assessment of the different models defined in Table 5.1. Each boxplot contains 25 values of negative log-likelihood function.

5.4 Discussion

The focus of this study was on the comparison of three modeling techniques for estimating the real forest situation of the beeches in the district of "Rachel-Lusen," National Park "Bayerischer Wald." We specified a structured additive model which accounts for the trees' variation in height, as well as for spatial and spatio-temporal effects. The objective was to estimate a surface representing the browsing probabilities on young beech trees within the borders of the "Rachel-Lusen" district. We provided a boosted version of the GAM model, i.e., the GAMBoost model, which succeeded in outperforming the classical GAM model in terms of stronger minimization of the out-of-sample risk.

We found that the spatial component does contribute to the fit considerably. The same holds for tree height, which should be considered when estimating the browsing probability in regeneration areas.

The assumption of a smooth relationship between the response and the covariates did not prove to be the most credible one among our model choices. A simple recursive partitioning of the geographical component via boosting with regression trees, i.e., the TreeBoost model, proved to obtain by far the smallest out-of-sample risk. This is mostly due to the irregular distribution of the regeneration areas leading to abrupt changes of the browsing pressure, especially in the populated regions. In addition, the TreeBoost model required less computational time.

Chapter 6

Summary and Conclusion

In this dissertation, we showed how boosting can be used to detect and estimate the impact of relevant factors on the dynamics of the mean and of the volatility in autoregressive time series. Specifically, we used componentwise boosting, which is designed for sorting out irrelevant (lagged) predictors. In the applications, we focused on multi-period ahead forecasts of the conditional mean and variance. In the simulation studies, we showed its interpretative potential by recovering the true dynamics of the simulated stochastic processes. In addition, boosting was also applied to a spatio-temporal dataset originating from outside the econometric field. In summary, we found the following strengths of boosting.

1. Evidently, boosting can be very competitive when estimating the conditional mean and variance of nonlinear high-order autoregressive time series. In a simulation study, it was superior to several alternative nonparametric methods in terms of goodness-of-fit.

2. Forecasting the monthly returns of German industrial production was most successfully carried out via componentwise boosting with linear base learners. This strategy was compared to the benchmark in macroeconomic forecasting, namely, the linear (vector) autoregressive model. Moreover, the boosting model was almost immune to the addition of potentially noninformative variables and their long history.

3. Forecasting the monthly volatility in the four broad asset classes of stocks, commodities, bonds, and foreign exchange, by boosting leads to more robust, i.e., less outlier prone prediction MSEs than does the GARCH benchmark. Our

boosting approach with regression-tree base learners performed very favorably for stocks and commodities relative to the common GARCH(1,1) benchmark model. The advantages are particularly convincing for longer forecasting horizons.

4. Our empirical results give insight into the “anatomy” of the volatility by identifying small groups of influential drivers. We found, indeed, highly nonlinear relationships between financial drivers and the volatility in stocks and commodities. This contrasts with the existing literature, which has almost exclusively concentrated on linear volatility dynamics.

In Chapter 2, we mentioned several inherent problems of boosting, e.g., lack of inference, estimation bias, false detections, and a computationally costly stopping condition. The existing solutions seem to have a varying success and further research is desirable. Furthermore, even though boosting is generally useful for fitting correlated data, these problems are even more severe for dependent covariates. The serial dependence in time series data might mislead the fitting procedure into producing erroneous transformations. Therefore, care must be taken in using boosting algorithms in time series with strong serial correlation.

Simultaneously modeling the conditional mean and variance in a multi-parameter framework, similarly to Schmid et al. (2010) and Mayr et al. (2012), is surely worth consideration. The increase in flexibility, however, raises some of the inherent problems beyond feasibility, e.g., the stopping condition can potentially get too imprecise if it should remain computationally solvable.

With regard to the applications of boosting, we showed that it helps to improve volatility forecasts. The next step is to use these forecasts in an ongoing project for option valuation and portfolio optimization and to check whether or not superior results can be achieved.

Appendix A

The Choice of Leading Indicators

The Ifo Business Climate Index is based on about 7,000 monthly survey responses of firms in manufacturing, construction, wholesaling and retailing. The firms are asked to give their assessments of the current business situation and their expectations for the next six months. The balance value of the current business situation is the percentage difference between "good" and "poor" responses; the balance value of the expectations is the percentage difference between "more favourable" and "more unfavourable" responses. The business climate is a transformed mean of both. For further information see Goldrian (2007). This index was used for forecasting German IP in Breitung and Jagodzinski (2001); Fritsche and Stephan (2002); Hüfner and Schröder (2002); Dreger and Schumacher (2005) among others.

The ZEW Indicator of Economic Sentiment is published monthly. Up to 350 financial experts take part in the survey. The indicator reflects the difference between the share of analysts that are optimistic and the share of analysts that are pessimistic with regard to the expected economic development in Germany within six months. For further details and for an application of forecasting IP in terms of a bivariate VAR model see Hüfner and Schröder (2002); Benner and Meier (2004); Dreger and Schumacher (2005).

The FAZ indicator (Frankfurter Allgemeine Zeitung) pools survey data and macroeconomic time series. It consists of the Ifo index (0.13), new orders in manufacturing industries (0.56), the real effective exchange rate of the Euro (0.06), the interest rate spread (0.08), the stock market index DAX (0.01), the number of job vacancies (0.05) and lagged industrial production (0.11). The Ifo index, orders in manufacturing and the number of job vacancies enter the indicator equation in levels, while the other variables are measured in first differences. FAZ was used for forecasting IP in

Breitung and Jagodzinski (2001), Dreger and Schumacher (2005).

The Early Bird indicator, compiled by Commerzbank, also pools different time series and stresses the importance of international business cycles for the German economy. Its components are the real effective exchange rate of the Euro (0.35), the short-term real interest rate (0.4), defined as the difference between the short-term nominal rate and core inflation, and the purchasing manager index of U.S. manufactures (0.25).

The OECD composite leading indicator is delivered by using a modified version of the Phase-Average Trend method (PAT) developed by the US National Bureau of Economic Research (NBER). The indicator is compiled by combining de-trended component series in either their seasonally adjusted or raw form. The component series are selected based on various criteria such as economic significance, cyclical behaviour, data quality, timeliness and availability. For Germany the following time series are compiled: Orders inflow or demand: tendency (manufacturing) (% balance), Ifo Business climate indicator (manufacturing) (% balance), Spread of interest rates (% annual rate), Total new orders (manufacturing), Finished goods stocks: level (manufacturing) (% balance) and Export order books: level (manufacturing) (% balance).

Financial indicators, such as overnight interbank interest rate an interest spread, are used as possible predictors as well. Stock and Watson (2003) have conducted a thorough case study for different OECD countries by forecasting Gross Domestic Product (GDP), Inflation and Industrial production. The information on the growth of the employment in Germany was taken from their paper.

Finally, a factor indicator obtained from a large data set from Germany, is included. The data set contains the German quarterly GDP and 111 monthly indicators from 1992 to 2006.[1]

[1] The estimated factor was provided by Christian Schumacher and is based on Marcellino and Schumacher (2007).

Bibliography

Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in *2nd International Symposium on Information Theory*, eds. Petrov, B. N. and Csaki, F., pp. 267–281. (page 23).

Ammer, C. (1996), "Impact of Ungulates on Structure and Dynamics of Natural Regeneration of Mixed Mountain Forests in the Bavarian Alps," *Forest Ecology and Management*, 88, 43–53. (page 84).

Andersen, T. G., Bollerslev, T., Christoffersen, P. F., and Diebold, F. X. (2006), "Volatility and Correlation Forecasting," in *Handbook of Economic Forecasting*, eds. Elliott, G., Granger, C., and Timmermann, A., Elsevier, vol. 1, chap. 15, pp. 777–878. (pages 56, 68).

Audrino, F. (2010), "What Drives Short Rate Dynamics? A Functional Gradient Descent Approach," Working Paper 640, University of St. Gallen. (page 43).

Audrino, F. and Barone-Adesi, G. (2006), "A Dynamic Model of Expected Bond Returns: A Functional Gradient Descent Approach," *Computational Statistics & Data Analysis*, 51, 2267–2277. (page 32).

Audrino, F. and Bühlmann, P. (2003), "Volatility Estimation with Functional Gradient Descent for Very Highdimensional Financial Time Series," *Journal of Computational Finance*, 6, 65–89. (pages 32, 57).

— (2009), "Splines for Financial Volatility," *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 71, 655–670. (pages 32, 57).

Augustin, N. H., Musio, M., von Wilpert, K., Kublin, E., Wood, S. N., and Schumacher, M. (2009), "Modeling Spatiotemporal Forest Health Monitoring Data," *Journal of the American Statistical Association*, 104, 899–911. (page 91).

Bai, J. and Ng, S. (2009), "Boosting Diffusion Indices," *Journal of Applied Econometrics*, 24, 607–629. (page 57).

Benner, J. and Meier, C. (2004), "Prognosegüte alternativer Frühindikatoren für die Konjunktur in Deutschland," *Jahrbücher für Nationalökonomie und Statistik*, 224, 637–652. (pages 43, 103).

Bollerslev, T. (1986), "Generalized Autoregressive Conditional Heteroskedasticity," *Journal of Econometrics*, 31, 307–327. (pages 4, 56).

Breiman, L. (1996), "Bagging Predictors," *Machine Learning*, 24, 123–140. (pages 13, 25, 60).

— (1998), "Arcing Classifiers," *The Annals of Statistics*, 26, 801–824. (page 10).

— (1999), "Prediction Games and Arcing Algorithms," *Neural Computation*, 11, 1493–1517. (page 10).

— (2001a), "Random Forests," *Machine Learning*, 45, 5–32. (page 29).

— (2001b), "Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)," *Statistical Science*, 16, 199–231. (page 1).

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), "Classification and Regression Trees," *Wadsworth, Belmont, CA, USA*. (pages 9, 59, 90).

Breitung, J. and Jagodzinski, D. (2001), "Prognoseeigenschaften alternativer Indikatoren der konjunkturellen Entwicklung in Deutschland," *Konjunkturpolitik*, 47, 292–314. (pages 103, 104).

Bühlmann, P. (2006), "Boosting for High-Dimensional Linear Models," *The Annals of Statistics*, 34, 559–583. (pages 23, 27, 35).

Bühlmann, P. and Hothorn, T. (2007a), "Boosting Algorithms: Regularization, Prediction and Model Fitting," *Statistical Science*, 22, 477–505, with discussion. (pages 9, 10, 13, 20, 23, 32, 35, 57).

— (2007b), "Rejoinder: Boosting Algorithms: Regularization, Prediction and Model Fitting," *Statistical Science*, 22, 516–522. (page 23).

Bühlmann, P. and Hothorn, T. (2010), "Twin Boosting: Improved Feature Selection and Prediction," *Statistics and Computing*, 20, 119–138. (page 26).

Bühlmann, P. and McNeil, A. J. (2002), "An Algorithm for Nonparametric GARCH Modelling," *Computational Statistics & Data Analysis*, 40, 665–683. (page 57).

Bühlmann, P. and van De Geer, S. (2011), *Statistics for High-Dimensional Data: Methods, Theory and Applications*, Springer-Verlag, New York. (pages 9, 24, 27).

Bühlmann, P. and Yu, B. (2003), "Boosting with the L_2 Loss: Regression and Classification," *Journal of the American Statistical Association*, 98, 324–339. (pages 10, 15, 22, 35, 57, 59, 88).

— (2006), "Sparse Boosting," *Journal of Machine Learning Research*, 7, 1001–1024. (page 23).

Campbell, J. Y. and Thompson, S. B. (2008), "Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average?" *Review of Financial Studies*, 21, 1509–1531. (page 71).

Cardarelli, R., Elekdag, S., and Lall, S. (2009), "Financial Stress, Downturns, and Recoveries," IMF Working Paper WP/09/100, International Monetary Fund. (pages 66, 76).

Chan, K. S. and Tong, H. (1986), "On Estimating Thresholds in Autoregressive Models," *Journal of Time Series Analysis*, 7, 179–190. (page 3).

Chen, R. and Tsay, R. S. (1993), "Nonlinear Additive ARX Models," *Journal of the American Statistical Association*, 88, 955–967. (pages 3, 32, 33, 37).

Chevillon, G. and Hendry, D. F. (2005), "Non-Parametric Direct Multi-Step Estimation for Forecasting Economic Processes," *International Journal of Forecasting*, 21, 201–218. (page 43).

Christiansen, C., Schmeling, M., and Schrimpf, A. (2012), "A Comprehensive Look at Financial Volatility Prediction by Economic Variables," *Journal of Applied Econometrics*, 27, 956–977. (pages 56, 57, 70).

Claveria, O., Pons, E., and Ramos, R. (2007), "Business and Consumer Expectations and Macroeconomic Forecasts," *International Journal of Forecasting*, 23, 47–69. (page 43).

Clements, M. P., Franses, P. H., and Swanson, N. R. (2004), "Forecasting Economic and Financial Time-Series with Non-Linear Models," *International Journal of Forecasting*, 20, 169–183. (page 43).

Cochrane, J. H. and Piazzesi, M. (2005), "Bond Risk Premia," *American Economic Review*, 95, 138–160. (pages 56, 66, 67).

Diebold, F. X. and Mariano, R. S. (1995a), "Comparing Predictive Accuracy," *Journal of Business & Economic Statistics*, 13, 253–265. (page 45).

— (1995b), "Comparing Predictive Accuracy," *Journal of Business & Economic Statistics*, 13, 253–263. (pages 71, 72).

Dreger, C. and Schumacher, C. (2005), "Out-of-sample Performance of Leading Indicators for the German Business Cycle. Single vs Combined Forecasts," *Journal of Business Cycle Measurement and Analysis*, 2, 71–88. (pages 43, 46, 103, 104).

Efron, B. (1979), "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics*, 7, 1–26. (page 89).

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 407–499, with discussion, and a rejoinder by the authors. (pages 24, 26).

Eiberle, K. (1989), "Über den Einfluss des Wildverbisses auf die Mortalität von jungen Waldbäumen in der oberen Montanstufe," *Schweizer Zeitschrift für Forstwesen*, 140, 1031–1042. (page 84).

Eiberle, K. and Nigg, H. (1987), "Grundlagen zur Beurteilung des Wildverbisses im Gebirgswald," *Schweizer Zeitschrift für Forstwesen*, 138, 474–785. (page 84).

Eilers, P. H. C. and Marx, B. D. (1996), "Flexible Smoothing with B-Splines and Penalties," *Statistical Science*, 11, 89–121. (pages 17, 33, 36, 87).

Elliot, G. and Timmermann, A. (2008), "Economic Forecasting," *Journal of Economic Literature*, 66, 3–56. (page 43).

Engle, R. F. (1982), "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation," *Econometrica*, 50, 987–1007. (pages 4, 56).

Engle, R. F., Ghysels, E., and Sohn, B. (2008), "On the Economic Sources of Stock Market Volatility," Working paper series, AFA 2008 New Orleans Meetings Paper. (page 56).

Fahrmeir, L., Kneib, T., and Lang, S. (2004), "Penalized Structured Additive Regression for Space-Time Data: A Bayesian Perspective," *Statistica Sinica*, 14, 731–761. (pages 3, 85).

Fan, J. and Li, R. (2001), "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties," *Journal of the American Statistical Association*, 96, 1348–1360. (page 25).

Fan, J. and Lv, J. (2008), "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 849–911. (page 25).

— (2010), "A Selective Overview of Variable Selection in High Dimensional Feature Space," *Statistica Sinica*, 20, 101–148. (pages 25, 27).

Fenske, N., Kneib, T., and Hothorn, T. (2011), "Identifying Risk Factors for Severe Childhood Malnutrition by Boosting Additive Quantile Regression," *Journal of the American Statistical Association*, 106, 494–510. (pages 19, 53).

Forstliches Gutachten (2006), *Forstliche Gutachten zur Situation der Waldverjungung 2006*, Bayerische Staatsministerium für Landwirtschaft und Forsten. (page 84).

French, K. R., Schwert, G. W., and Stambaugh, R. F. (1987), "Expected Stock Returns and Volatility," *Journal of Financial Economics*, 19, 3–29. (page 68).

Freund, Y. and Schapire, R. (1996), "Experiments With a New Boosting Algorithm," in *Proceedings of the Thirteenth International Conference on Machine Learning Theory*, San Francisco: Morgan Kaufmann, pp. 148–156. (pages 1, 9, 21).

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 19, 1–67. (pages 3, 33).

— (2001), "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, 29, 1189–1232. (pages 1, 10, 14, 18, 34, 35, 60).

Friedman, J. H., Hastie, T., and Tibshirani, R. (2000), "Additive Logistic Regression: A Statistical View of Boosting," *The Annals of Statistics*, 28, 337–407, with discussion. (pages 1, 9).

Fritsche, U. and Stephan, S. (2002), "Leading Indicators of German Business Cycles – An Assessment of Properties," *Jahrbücher für Nationalökonomie und Statistik*, 222, 289–311. (page 103).

Fritz, P. (2006), *Ökologischer Waldumbau in Deutschland - Fragen, Antworten, Perspektiven*, Oekom, Munich. (page 83).

Gertheiss, J. and Tutz, G. (2009), "Penalized Regression with Ordinal Predictors," *International Statistical Review*, 77, 345–365. (page 17).

Goldrian, G. (2007), *Handbook of Survey-Based Business Cycle Analysis*, Edward Elgar. (page 103).

Goyal, A. and Welch, I. (2003), "Predicting the Equity Premium with Dividend Ratios," *Management Science*, 49, 639–654. (page 56).

Granger, C. W. J. and Andersen, A. P. (1978), *An Introduction to Bilinear Time Series Models*, Göttingen. (page 3).

Hamilton, J. D. (1989), "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle," *Econometrica*, 57, 357–384. (page 3).

— (1994), *Time Series Analysis*, Princeton University Press. (page 32).

Hansen, M. H. and Yu, B. (2001), "Model Selection and the Principle of Minimum Description Length," *Journal of the American Statistical Association*, 96, 746–774. (page 23).

Hansen, P. R., Lunde, A., and Nason, J. M. (2010), "The Model Confidence Set," CREATES Research Papers 2010-76, School of Economics and Management, University of Aarhus. (page 52).

Harvey, D., Leybourne, S., and Newbold, P. (1997), "Testing the Equality of Prediction Mean Squared Errors," *International Journal of Forecasting*, 13, 281–291. (pages 45, 51, 71).

Hastie, T. (2007), "Comment: Boosting Algorithms: Regularization, Prediction and Model Fitting," *Statistical Science*, 22, 513–515. (page 23).

Hastie, T. and Tibshirani, R. (1990), *Generalized Additive Models*, Chapman & Hall. (pages 11, 23, 26, 33, 91).

Hastie, T., Tibshirani, R., and Friedman, J. (2009a), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer-Verlag, Berlin, 2nd ed. (pages 9, 10, 14, 23, 27, 33, 34, 81).

Hastie, T., Tibshirani, R., Leisch, F., Hornik, K., and Ripley, B. D. (2009b), *mda: Mixture and Flexible Discriminant Analysis*, R package version 0.3-4. (page 54).

Hoerl, A. E. and Kennard, R. W. (1970), "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, 12, 55–67. (page 17).

Hofner, B. (2012), *Boosting in Structured Additive Models*, Verlag Dr. Hut, Munich. (page 28).

Hofner, B., Hothorn, T., Kneib, T., and Schmid, M. (2011), "A Framework for Unbiased Model Selection Based on Boosting," *Journal of Computational and Graphical Statistics*, 20, 956–971. (page 16).

Hofner, B., Mayr, A., Robinzonov, N., and Schmid, M. (2012), "Model-based Boosting in R: A Hands-on Tutorial Using the R Package mboost," *Computational Statistics*, 1–33. (pages 6, 16, 28).

Hothorn, T., Buehlmann, P., Kneib, T., Schmid, M., and Hofner, B. (2009), *mboost: Model-Based Boosting*, R package, version 1.0-7. (page 53).

— (2011), *Model-Based Boosting*, R package version 2.1-1. (page 59).

Hothorn, T., Hornik, K., and Zeileis, A. (2006), "Unbiased Recursive Partitioning: A Conditional Inference Framework," *Journal of Computational and Graphical Statistics*, 15, 651–674. (pages 59, 60, 62, 90).

Huang, J., Lu, L., and Wu, B. (2011), "Macro Factors and Term Structure of Treasury Bond Volatility," Working Paper WP/09/100. (page 78).

Huang, J. Z. and Yang, L. (2004), "Identification of Non-Linear Additive Autoregressive Models," *Journal of the Royal Statistical Society Series B(Statistical Methodology)*, 66, 463–477. (pages 32, 33, 37, 44, 54).

Hüfner, F. P. and Schröder, M. (2002), "Prognosegehalt von ifo-Geschäftserwartungen und ZEW-Konjunkturerwartungen: Ein ökonometrischer Vergleich," *Jahrbücher für Nationalökonomie und Statistik*, 222, 316–336. (pages 43, 103).

Hurvich, C. M., Simonoff, J. S., and Tsai, C.-L. (1998), "Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion," *Journal of the Royal Statististical Society, Series B*, 60, 271–293. (page 23).

Hyndman, R. J. and Koehler, A. B. (2006), “Another Look at Measures of Forecast Accuracy,” *International Journal of Forecasting*, 22, 679–688. (page 43).

Jorion, P. (1995), “Predicting Volatility in the Foreign Exchange Market,” *Journal of Finance*, 50, 507–28. (page 74).

Kearns, M. and Valiant, L. (1994), “Cryptographic Limitations on Learning Boolean Formulae and Finite Automata,” *Journal of the Association for Computing Machinery*, 41, 67–95. (page 9).

Kneib, T., Hothorn, T., and Tutz, G. (2009), “Variable Selection and Model Choice in Geoadditive Regression Models,” *Biometrics*, 65, 626–634. (pages 15, 28, 53, 86, 87).

Knoke, T., Ammer, C., Stimm, B., and Mosandl, R. (2008), “Admixing Broad-Leaved to Coniferous Tree Species–A Review on Yield, Ecological Stability and Economics,” *European Journal of Forest Research*, 127, 89–101. (page 83).

Knoke, T. and Seifert, T. (2008), “Integrating Selected Ecological Effects of Mixed European Beech–Norway Spruce Stands in Bioeconomic Modelling,” *Ecological Modelling*, 210, 487–498. (page 83).

Koenker, R. (2005), *Quantile Regression*, Cambridge University Press. (page 19).

Kuester, K., Mittnik, S., and Paolella, M. S. (2006), “Value-at-Risk Prediction: A Comparison of Alternative Strategies,” *Journal of Financial Econometrics*, 4, 53–89. (page 56).

Leeb, H. and Pötscher, B. M. (2005), “Model Selection and Inference: Facts and Fiction,” *Econometric Theory*, 21, 21–59. (pages 25, 27).

Lewis, P. and Stevens, J. G. (1991), “Nonlinear Modeling of Time Series Using Multivariate Adaptive Regression Splines (MARS).” *Journal of the American Statistical Association*, 86, 864–877. (pages 3, 32, 33).

Lunde, A. and Hansen, P. R. (2005), “A forecast comparison of volatility models: does anything beat a GARCH(1,1)?” *Journal of Applied Econometrics*, 20, 873–889. (page 70).

Lustig, H., Roussanov, N., and Verdelhan, A. (2011), “Common Risk Factors in Currency Markets,” *Review of Financial Studies*, 24, 3731–3777. (pages 56, 66, 67).

Lütkepohl, H. (1991), *Introduction to Multiple Time Series Analysis*, Springer, Berlin. (page 32).

— (2006), *New Introduction to Multiple Time Series Analysis*, Springer. (page 32).

Lutz, R. W., Kalisch, M., and Bühlmann, P. (2008), "Robustified L_2 Boosting," *Computational Statistics & Data Analysis*, 52, 3331–3341. (pages 35, 53).

Marcellino, M. and Schumacher, C. (2007), "Factor Nowcasting of German GDP With Ragged-Edge Data. A Model Comparison Using MIDAS Projections," Tech. rep., Bundesbank Discussion Paper, Series 1, 34/2007. (page 104).

Marcellino, M., Stock, J. H., and Watson, M. W. (2006), "A Comparison of Direct and Iterated Multistep AR Methods for Forecasting Macroeconomic Time Series," *Journal of Econometrics*, 135, 499–526. (page 43).

Mason, L., Baxter, J., Bartlett, P. L., and Frean, M. (2000), "Functional Gradient Techniques for Combining Hypotheses," in *Advances in Large Margin Classifiers*, eds. Smola, A. J., Bartlett, P. L., Schölkopf, B., and Schuurmans, D., MIT Press, pp. 221–246. (page 10).

Matías, J. M., Febrero-Bande, M., González-Manteiga, W., and Reboredo, J. C. (2010), "Boosting GARCH and Neural Networks for the Prediction of Heteroskedastic Time Series," *Mathematical and Computer Modelling*, 51, 256–271. (pages 32, 57).

Mayr, A., Fenske, N., Hofner, B., Kneib, T., and Schmid, M. (2012), "Generalized Additive Models for Location, Scale and Shape for High Dimensional Data—A Flexible Approach Based on Boosting," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61, 403–427. (pages 22, 102).

McCullagh, P. and Nelder, J. (1989), *Generalized Linear Models*, Chapman & Hall/CRC. (page 11).

Meinshausen, N. and Bühlmann, P. (2010), "Stability Selection," *Journal of the Royal Statistical Society, Series B*, 72, 417–473. (pages 24, 25, 26).

Menkhoff, L., Sarno, L., Schmeling, M., and Schrimpf, A. (2011), "Carry Trades and Global Foreign Exchange Volatility," CEPR Discussion Papers 8291, C.E.P.R. Discussion Papers. (pages 66, 67).

Mittnik, S., Robinzonov, N., and Spindler, M. (2012), "Boosting the Anatomy of Volatility," Tech. Rep. 124, Ludwig-Maximilians-Universität München. (pages 7, 20, 55).

Moog, M. (2008), *Bewertung von Wildschäden im Wald*, Neumann-Neudamm, Melsungen. (page 85).

Motta, R. (2003), "Ungulate Impact on Rowan (*Sorbus aucuparia* L) and Norway spruce (*Picea abies* (L) Karst) Height Structure in Mountain Forests in the Italian Alps," *Forest Ecology and Management*, 181, 139–150. (page 84).

Nelson, D. B. (1991), "Conditional Heteroskedasticity in Asset Returns: A New Approach," *Econometrica*, 59, 347–370. (page 58).

Nowak, S. and Treepongkaruna, S. (2008), "Modeling and Forecasting Volatility in Foreign Exchange Markets," Working Paper Series in Finance FINM0042WP, School of Finance and Applied Statistics, Australian National University. (page 74).

Pastor, L. and Stambaugh, R. F. (2003), "Liquidity Risk and Expected Stock Returns," *Journal of Political Economy*, 111, 642–685. (pages 66, 67).

Patton, A. J. (2011), "Volatility Forecast Comparison Using Imperfect Volatility Proxies," *Journal of Econometrics*, 160, 246–256. (page 69).

Paye, B. S. (2012), "'Déjà Vol': Predictive Regressions for Aggregate Stock Market Volatility Using Macroeconomic Variables," *Journal of Financial Economics*, in Press. (page 56).

Pfaff, B. (2008), "VAR, SVAR and SVEC Models: Implementation Within R Package vars," *Journal of Statistical Software*, 27, 1–32. (page 54).

Prien, S. (1997), *Wildschäden im Wald*, Paul Parey, Berlin. (page 85).

R Development Core Team (2009), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0. (page 53).

— (2012), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0. (page 59).

Rigby, R. A. and Stasinopoulos, D. M. (2005), "Generalized Additive Models for Location, Scale and Shape," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54, 507–554. (page 22).

Robinzonov, N. and Hothorn, T. (2010), "Boosting for Estimating Spatially Structured Additive Models," in *Statistical Modelling and Regression Structures. Festschrift in Honour of Ludwig Fahrmeir*, Physica-Verlag, Heidelberg, pp. 181–196. (pages 7, 20, 83).

Robinzonov, N., Tutz, G., and Hothorn, T. (2012), "Boosting Techniques for Nonlinear Time Series Models," *AStA Advances in Statistical Analysis*, 96, 99–122. (pages 7, 31, 69).

Robinzonov, N. and Wohlrabe, K. (2010), "Freedom of Choice in Macroeconomic Forecasting," *CESifo Economic Studies*, 56, 192–220. (page 43).

Rosset, S., Zhu, J., and Hastie, T. (2004), "Boosting as a Regularized Path to a Maximum Margin Classifier," *The Journal of Machine Learning Research*, 5, 941–973. (page 10).

Rüegg, D. (1999), "Zur Erhebung des Einflusses von Wildtieren auf die Waldverüngung," *Schweizer Zeitschrift für Forstwesen*, 150, 327–331. (page 85).

Schapire, R., Freund, Y., Bartlett, P., and Lee, W. (1998), "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *The Annals of Statistics*, 26, 1651–1686. (page 9).

Schmid, M. and Hothorn, T. (2008), "Boosting Additive Models Using Component-Wise P-Splines," *Computational Statistics & Data Analysis*, 53, 298–311. (pages 37, 88).

Schmid, M., Potapov, S., Pfahlberg, A., and Hothorn, T. (2010), "Estimation and Regularization Techniques for Regression Models with Multidimensional Prediction Functions," *Statistics and Computing*, 20, 139–150. (pages 22, 102).

Schwert, G. W. (1989), "Why Does Stock Market Volatility Change over Time?" *Journal of Finance*, 44, 1115–53. (pages 56, 68).

Shafik, N. and Tutz, G. (2009), "Boosting Nonlinear Additive Autoregressive Time Series," *Computational Statistics & Data Analysis*, 53, 2453–2464. (pages 37, 53).

Stock, J. H. and Watson, M. W. (2003), "Forecasting Output and Inflation: The Role of Asset Prices," *Journal of Economic Literature*, 41, 788–829. (page 104).

Strasser, H. and Weber, C. (1999), "On the Asymptotic Theory of Permutation Statistics," *Mathematical Methods of Statistics*, 8, 220–250. (page 62).

Teräsvirta, T., van Dijk, D., and Medeiros, M. C. (2005), "Linear Models, Smooth Transition Autoregressions, and Neural Networks for Forecasting Macroeconomic Time Series: A Reexamination," *International Journal of Forecasting*, 21, 755–774. (page 43).

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, 58, 267–288. (page 24).

Tong, H. (1978), "On a Threshold Model," in *Pattern Recognition and Signal Processing*, ed. Chen, C., Sijthoff & Noordhoff, Netherlands, pp. 575–586. (page 3).

Tsay, R. S. (2005), *Analysis of Financial Time Series*, Wiley, New York, 2nd ed. (page 2).

Tschernig, R. and Yang, L. (2000), "Nonparametric Lag Selection for Time Series," *Journal of Time Series Analysis*, 21, 457–487. (page 37).

Tutz, G. and Binder, H. (2006), "Generalized Additive Modelling With Implicit Variable Selection by Likelihood Based Boosting," *Biometrics*, 62, 961–971. (page 10).

Viceira, L. M. (2012), "Bond Risk, Bond Return Volatility, and the Term Structure of Interest Rates," *International Journal of Forecasting*, 28, 97–117. (page 78).

Weisberg, P., Bonavia, F., and Bugmann, H. (2005), "Modeling the Interacting Effects of Browsing and Shading on Mountain Forest Regeneration (*Picea abies*)," *Ecological Modelling*, 185, 213–230. (page 84).

Weisberg, S. (1980), *Applied Linear Regression*, Wiley, New York. (page 26).

Welch, I. and Goyal, A. (2008), "A Comprehensive Look at The Empirical Performance of Equity Premium Prediction," *Review of Financial Studies*, 21, 1455–1508. (pages 56, 66).

Wood, S. (2009), *mgcv: GAMs with GCV smoothness estimation and GAMMs by REML/PQL*, R package, version 1.4-1.1. (page 54).

Wood, S. N. (2006), *Generalized Additive Models: An Introduction with R*, Chapman & Hall/CRC. (pages 54, 87).

Zhang, T. and Yu, B. (2005), "Boosting with Early Stopping: Convergence and Consistency," *The Annals of Statistics*, 33, 1538–1579. (page 27).

Zhao, P. and Yu, B. (2007), "Stagewise Lasso," *The Journal of Machine Learning Research*, 8, 2701–2726. (pages 13, 24, 26, 27).

Zou, H. (2006), "The Adaptive Lasso and Its Oracle Properties," *Journal of the American Statistical Association*, 101, 1418–1429. (page 26).

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12.07.11, § 8, Abs. 2 Pkt. .5.)

Hiermit erkläre ich an Eidesstatt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

Name, Vorname

Ort, Datum

Unterschrift Doktorand/in

Formular 3.2