Extending the Boosting Framework based on Bayesian Methodology

DER WIRTSCHAFTSWISSENSCHAFTLICHE FAKULTÄT
DER GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN
ZUR
ERLANGUNG DES DOKTORGRADES

DOCTOR RERUM POLITICARUM (DR. RER. POL.)

VORGELEGT VON
BOYAO ZHANG
AUS: HEILONGJIANG

Georg-August-Universität Göttingen
Professur für Raumbezogene Datenanalyse und Statistische Lernverfahren

Als Dissertation genehmigt
von der Wirtschaftswissenschaftlichen Fakultät
der Georg-August-Universität Göttingen

Tag der mündlichen Prüfung:	24. Januar 2023
Vorsitzender des Promotionsorgans:	Prof. Dr. Elisabeth Bergherr
Gutachter:	Prof. Dr. Thomas Kneib
	Dr. Tobias Hepp

Abstract

The boosting technique emerged from machine learning has become a widely used method to estimate statistical models. As one of the most successful variants, componentwise gradient boosting has been favored by more and more statisticians since its iterative procedure not only provides intuitive variable selection in high-dimensional analysis, but also supplies additional flexibility to estimate various types of additive regression terms. But its dogmatic estimates, i.e. its direct and unquestionable estimation conclusion, do not deliver any information about the error risk of estimation and prediction, which, however, is the basis for many statistical analyses.

As one of the most essential conventional statistical theories, Bayesian methodology maintains the ability to quantify uncertainty. Due to its unique prior philosophy, it has grown immensely in the past decades and has led to the development of innumerable new models. However, it often fails to give precise and unambiguous guidelines for the variable selection, which in turn is the advantage of boosting.

This thesis proposes a Bayesian-based boosting theory, which integrates Bayesian inference in the boosting framework. Componentwise boosting guarantees the high-dimensional analysis and the flexibility of base-learners since additive terms are updated individually. Furthermore, each base-learner inferred by Bayesian inference also preserves additional Bayesian properties such as the prior and the credible-based uncertainty quantification. The proposed Bayesian-based boosting method combines the strengths of the two approaches and overcomes the weaknesses of both.

This thesis firstly solves the problem of imbalanced updates of predictors in generalized additive models for location, scale and shape (GAMLSS) estimated using gradient boosting by introducing the adaptive step-length. Then, through the implementation of Bayesian learners in the gradient boosting framework for linear mixed models (LMM), the validity of the combination of Bayesian and boosting concepts is preliminarily verified. The complete Bayesian-based boosting framework is eventually presented by applying it to a generalized model family, namely structured additive regression (STAR) models.

Overall, the proposed Bayesian-based boosting is not only the first systematic study of the fusion of Bayesian inference and boosting techniques, but also an attempt to integrate machine learning and statistics at a deeper level.

Zusammenfassung

Die aus dem maschinellen Lernen hervorgegangene statistische Boostingtechnik ist zu einer weit verbreiteten Methode zur Schätzung statistischer Modelle geworden. Als eine der erfolgreichsten Varianten wird das komponentenweise Gradientenboosting von immer mehr Statistikern favorisiert, da das iteratives Verfahren nicht nur eine intuitive Variablenauswahl in der Analyse von hochdimensionalen Datensätzen ermöglicht, sondern auch zusätzliche Flexibilität bietet, um verschiedene Arten von additiven Regressionstermen zu schätzen. Da Boostingmodelle nur Punktschätzer liefern, lassen sich in der Regel keine Aussage über das Fehlerrisiko oder Vorhersage treffen, was jedoch die Grundlage vieler statistischer Analysen ist.

Als eine der wichtigsten konventionellen statistischen Theorien bewahrt die bayesianische Methodik die Fähigkeit, Unsicherheit zu quantifizieren. Aufgrund ihrer einzigartigen a priori Philosophie ist die Methodik in den letzten Jahrzehnten immens gewachsen und hat viele neue Modellarten hervorgebracht. Allerdings fehlt es oft an präzisen und eindeutigen Vorgaben für die Variablenauswahl, was wiederum der Vorteil des Boostings ist.

Diese Dissertation schlägt eine bayesianische Boostingtheorie vor, die die bayesianische Inferenz in den Rahmen von Boostingtechniken integriert. Das komponentenweise Boosting ermöglicht die hochdimensionalen und flexiblen Analysen von Basis-Lernern, da additive Terme einzeln aktualisiert werden. Zusätzlich behält jeder durch die bayesianische Inferenz abgeleitete Basis-Lerner weitere bayesianische Eigenschaften wie zum Beispiel die Prioritheorie und die glaubwürdige Unsicherheitsquantifizierung. Das vorgeschlagene bayesianische Boostingverfahren kombiniert also die Stärken und überwindet die Schwächen der beiden Ansätze.

Diese Arbeit löst zunächst das Problem unausgeglichener Aktualisierungen von Prädiktoren in verallgemeinerten additiven Modellen für Lage-, Skalen- und Formparameter (GAMLSS), die mit Hilfe des Gradientenboosting geschätzt werden, indem eine adaptive Schrittlänge eingeführt wird. Dann wird durch die Implementierung von bayesianischen Lernern im Rahmen des Gradientenboosting für lineare gemischte Modelle (LMM) die Validität der Kombination von bayesianischen und Boosting-Konzepten vorläufig verifiziert. Das vollständige bayesianische Boosting-Framework wird schließlich präsentiert, indem es auf eine verallgemeinerte Modellfamilie angewendet wird, nämlich auf strukturierte additive Regressionsmodelle (STAR).

Insgesamt ist das vorgeschlagene bayesianische Boosting nicht nur die erste systematische Studie zur Verschmelzung von bayesianischer Inferenz und Boostingtechniken, sondern auch ein Versuch, maschinelles Lernen und Statistik auf einer tieferen Ebene zu integrieren.

Contents

1 Introduction **1**

2 Adaptive step-length selection in gradient boosting for Gaussian location and scale models **7**
- 2.1 Boosted GAMLSS . . . 9
  - 2.1.1 GAMLSS and componentwise gradient boosting . . . 10
  - 2.1.2 Cyclical boosted GAMLSS . . . 13
  - 2.1.3 Non-cyclical boosted GAMLSS . . . 13
- 2.2 Adaptive Step-Length . . . 16
  - 2.2.1 Boosted GAMLSS with adaptive step-length . . . 16
  - 2.2.2 Gaussian location and scale models . . . 18
  - 2.2.3 (Semi-)Analytical adaptive step-length . . . 22
- 2.3 Simulation Study . . . 23
  - 2.3.1 Gaussian Location and Scale Model . . . 23
  - 2.3.2 Large Variance with resulting Imbalance between Location and Scale . . . 27
- 2.4 Applications . . . 32
  - 2.4.1 Malnutrition of children in India . . . 32
  - 2.4.2 Riboflavin dataset . . . 35
- 2.5 Conclusions and Outlook . . . 37

3 Bayesian Learners in Gradient Boosting for Linear Mixed Models **39**
- 3.1 Methods . . . 41
  - 3.1.1 Model specification . . . 42
  - 3.1.2 Bayesian boosting inference method . . . 43
  - 3.1.3 Stopping criterion . . . 51
- 3.2 Simulation . . . 53
  - 3.2.1 Estimation accuracy . . . 53
  - 3.2.2 Random effects selection . . . 57
  - 3.2.3 Performance of uncertainty estimation . . . 60
- 3.3 Application . . . 63
- 3.4 Discussion and Outlook . . . 65

4 Bayesian-based Boosting for Quantifying Uncertainty in Structured Additive Regression **69**
- 4.1 Methods . . . 72
  - 4.1.1 Model specification . . . 73
  - 4.1.2 Bayesian penalized regression . . . 74
  - 4.1.3 Componentwise boosting . . . 75

4.1.4 Bayesian-based boosting . . . 76
4.1.5 Discussion of the uncertainty . . . 78
4.2 Simulation . . . 79
4.2.1 Model setup . . . 80
4.2.2 Linear regression . . . 81
4.2.3 Non-linear regression . . . 84
4.3 Application . . . 88
4.4 Summary and discussion . . . 90

5 Conclusion and Afterthoughts 93
5.1 Summary of the thesis . . . 93
5.2 Afterthoughts for further research . . . 94
5.2.1 Further investigating analytical adaptive step-lengths . . . 94
5.2.2 Dependence between the prior coefficients and uncertainty . . . 95
5.2.3 Complexity of Bayesian-based boosting . . . 96

A Appendix 1
A.1 Derive the analytical ASL for the Gaussian distribution . . . 1
A.1.1 Optimal step-length for $\boldsymbol{\mu}$ . . . 2
A.1.2 Optimal step-length for $\boldsymbol{\sigma}$ . . . 3
A.2 Additional simulation graphics . . . 4
A.3 Additional simulation table . . . 8
A.4 Estimated coefficients of riboflavin dataset . . . 8

Acknowledgement 11

Declaration 13

List of Figures

2.1 Boxplots of MSEs for FSL and ASL 25
2.2 Boxplot of false positives for FSL and ASL 26
2.3 Comparison of the optimal step-lengths $\nu^{*[m]}_{j^*,\boldsymbol{\mu}}$ and $\nu^{*[m]}_{j^*,\boldsymbol{\sigma}}$ in SAASL . . . 27
2.4 Distribution of coefficient estimates 29
2.5 Pairwise comparison of the estimated coefficients 30
2.6 Distribution of $p_{m_{\boldsymbol{\mu}}}$ 31
2.7 Optimal step-length of each model parameter 34

3.1 Boxplots of each model evaluation metric 57
3.2 Distribution of MCMC samples with respect to all individuals 58
3.3 Coefficients paths of fixed and random terms 60
3.4 Interval estimates of random effects 62
3.5 Coverage probabilities for each random effect 63
3.6 Densities of the mode for random effect $YXLD - at$ for each specimen 65

4.1 Convergence behavior of `bboost` model 83
4.2 Summarized standard deviation of estimates 84
4.3 Smooth effects estimated by `bboost` 85
4.4 Comparison of credible intervals between `bboost` and `BayesX` 86
4.5 Coverage rates of predictions between `bboost` and `BayesX` 87
4.6 Comparison of smooth effects between `bboost` and `mboost` 89
4.7 Spatial effects of the district variable on rents 90

A.1 Boxplot of the estimated coefficients of η_μ 5
A.2 Boxplot of the estimated coefficients of η_σ 6
A.3 Summary of the negative log-likelihood 7
A.4 Stopping iteration tuned by 10-fold CV 7

List of Tables

2.1 Summary of the in-sample MSE 29
2.2 Number of simulations with false positives and false negatives 31
2.3 Comparison of the estimated coefficients. 34
2.4 Number of chosen variables for $\eta_{\boldsymbol{\mu}}$ and $\eta_{\boldsymbol{\sigma}}$ 36
2.5 Comparison of the out-of-sample MSE 36

3.1 Model evaluation metric between `grbLMM` and `BayesBoost` 55
3.2 Model evaluation metric between `grbLMM` and `BayesBoost` 56
3.3 False positive rate (FPR) and false negative rate (FNR) under different noise-to-signal ratio (NSR) 56

4.1 Estimation accuracy of different methods 82

A.1 Average MSE of the estimated coefficients for both model parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ 8
A.2 Estimated coefficients of the $\boldsymbol{\mu}$-submodel 9
A.3 Estimated coefficients of $\boldsymbol{\sigma}$-submodel 10

Chapter 1

Introduction

Boosting is a machine learning ensemble technique in which a number of weak learners are combined to create a stronger one. The so-called weak learners are a series of models, each of which is capable of making predictions that are slightly better than a random guess, while the strong learners are able to make more accurate predictions than any of the individual weak learners. The idea of boosting was first introduced by Freund (1995) and Freund and Schapire (1996, 1997) propose the first applicable algorithm, AdaBoost, by building a committee of weak classifiers and weighting their predictions to form a single strong classifier. Since then, boosting has been used extensively in a variety of machine learning tasks and many different boosting algorithms have been developed. Nowadays, boosting is considered one of the most effective and widely-used ensemble learning techniques in machine learning.

The first breakthrough in boosting is the proposal of *gradient boosting* by Friedman et al. (2000). They show that gradient boosting can be used to create a strong predictive model by sequentially adding weak learners to the ensemble, each of which attempts to correct the mistakes of the previous learners, and it uses a gradient descent to optimize a loss function. Subsequently, Friedman (2001) proposes the concept of *model-based boosting*, which uses a model to guide the training process, i.e. let weak learners focus on the areas of the data where the model is less certain to improve the overall accuracy of the ensemble. The idea of model-based boosting enables boosting outcomes to have statistical properties. Thus, the era of statistical learning is ushered in.

One of the seminal works in statistical learning is the implementation of the *componentwise* concept in gradient boosting proposed by Bühlmann and Yu (2003). In the componentwise gradient boosting method, the variables are partitioned into several disjoint subset (usually only one covariate in a subset) and each weak learner is trained on each subset. This partition not only reduces the high-dimensional analysis, which is the weakness of conventional statistical approaches, to a simple regression problem, but also provides the flexibility to estimate various types of base-learners in one ensemble model. In recent years, almost all research on statistical boosting is established on the componentwise gradient boosting framework.

However, the estimation with boosting technique is usually dogmatic, that is it lacks straightforward ways to construct estimators for the precision of parameters such as variance or confidence intervals, which, nevertheless, is the basis of statistical analyses. The conventional statistical inference methods on the other side are able to quantify the uncertainty of estimates, which include both frequentist and Bayesian statistics, but they are not good at dealing with high-dimensional data. Even though regularization techniques such Lasso or ridge regression are available, the conventional approaches still very often fail to give precise and unambiguous guidelines for the selection or the exclusion of variables. The complementary relationship between boosting technique and conventional statistical inference on these points makes it natural to further integrate the two methods.

From the frequentist statistic perspective, Tutz and Binder (2006) propose a likelihood-based boosting framework, since the general estimation method used in gradient boosting is the least squares method, while in low-dimensional settings, another typical inference method is the maximum likelihood. In the special case of L_2 loss, likelihood-based boosting coincides with gradient boosting. Nevertheless, even though the paper indicates the possibility of constructing approximate confidence intervals in likelihood-based boosting, this point did not receive much attention until the publication of Rügamer and Greven (2020), which proposes inference for L_2-boosting. Compared to previous ad-hoc solutions such as permutation tests or bootstrapping, using a classical statistical method to quantify the uncertainty in boosting has various advantages.

In contrast to the relatively wider application of likelihood-based boosting, there is still little research on the implementation of Bayesian inference in boosting. Even though the Bayesian theorem was proposed in the eighteenth century, long before the fundamental theories proposed at the beginning of the twentieth century, which underpin modern statistics, its real development came with the rise of personal computers. Bayesian statistics has grown immensely in the last few decades and it has rendered an substantial amount of new types of models. The successful application of the boosting technique to the field of statistics makes it desirable to establish a Bayesian-based boosting framework, which is exactly the goal of this thesis.

Overall, the thesis mainly consists of three chapters, where each chapter represents an individual project. Chapter 2 introduces some basic concepts of gradient boosting and addresses the problem of imbalanced updates of predictors when applying it to complex models such as generalized additive models for location, scale and shape (GAMLSS) by introducing an adaptive step-length. The implementation of Bayesian inference in boosting is intensively discussed in Chapter 3 and 4. Chapter 3 firstly applies the Bayesian learner in gradient boosting to the specific linear mixed models (LMMs), which makes quantifying the uncertainty of random effects in boosting possible. Then, in Chapter 4, the flexible Bayesian-based boosting framework is proposed for the more general family, the structured additive regression (STAR) models, which cover not only linear and random effects as in LMMs, but also smooth and spatial learners. A short summary of each chapter is given in the following:

Chapter 2: Adaptive step-length selection in gradient boosting for Gaussian location and scale models

Tuning of model-based boosting algorithms relies mainly on the number of iterations, while the step-length is fixed at a predefined value. For complex models with several predictors such as Generalized Additive Models for Location, Scale and Shape (GAMLSS), imbalanced updates of predictors, where some distribution parameters are updated more frequently than others, can be a problem that prevents some submodels to be appropriately fitted within a limited number of boosting iterations. We propose an approach using adaptive step-length (ASL) determination within a non-cyclical boosting algorithm for Gaussian location and scale models, as an important special case of the wider class of GAMLSS, to prevent such imbalance. Moreover, we discuss properties of the ASL and derive a semi-analytical form of the ASL that avoids manual selection of the search interval and numerical optimization to find the optimal step-length, and consequently improves computational efficiency. We show competitive behavior of the proposed approaches compared to penalized maximum likelihood and boosting with a fixed step-length for Gaussian location and scale models in two simulations and two applications, in particular for cases of large variance and/or more variables than observations. In addition, the underlying concept of the ASL is also applicable to the whole GAMLSS framework and to other models with more than one predictor like zero-inflated count models, and brings up insights into the choice of the reasonable defaults for the step-length in the simpler special case of (Gaussian) additive models. This chapter is based on:

> **Zhang, B., Hepp, T., Greven, S., Bergherr, E.** (2022). Adaptive step-length selection in gradient boosting for Gaussian location and scale models. *Computational Statistics*, pages 1-38.

Chapter 3: Bayesian Learners in Gradient Boosting for Linear Mixed Models

Selection of relevant fixed and random effects without prior choices made from possibly insufficient theory is important in mixed models. Inference with current boosting techniques suffers from biased estimates of random effects and the inflexibility of random effects selection. This chapter proposes a new inference method `BayesBoost` that integrates a Bayesian learner into gradient boosting with simultaneous estimation and selection of fixed and random effects in linear mixed models. The method introduces a novel selection strategy for random effects, which allows for computationally fast selection of random slopes even in high-dimensional data structures. Additionally, the new method not only overcomes the shortcomings of Bayesian inference in giving precise and unambiguous guidelines for the selection of covariates by benefiting from boosting techniques, but also provides Bayesian ways to construct estimators for the precision of parameters such as variance components or credible intervals, which are not available in conventional boosting frameworks. The effectiveness of the new approach can be observed via simulation and in a real-world application. This chapter is based on:

> **Zhang, B., Griesbach, C., Bergherr, E.** (2022). Bayesian learners in gradient boosting for linear mixed models. *The International Journal of Biostatistics.*

Chapter 4: Bayesian-based Boosting for Quantifying Uncertainty in Structured Additive Regression

The boosting method is widely used in statistical learning, but its results are dogmatic, that is, it gives a direct and unquestionable estimation conclusion, which does not provide information about the error risk of estimation and prediction, i.e. the uncertainty of estimates, which is actually the basis for many statistical analyses. In this chapter, we propose a Bayesian-based boosting framework for structured additive regression models, which integrates Bayesian penalized inference into componentwise gradient boosting, enabling the novel method to specifically benefit from the uncertainty estimation of Bayesian inference and from the intuitive guidelines for the selection of variables of boosting techniques. The results of both linear and non-linear simulations indicate that the proposed method absorbs the advantages of both worlds well by maintaining a balance between estimation accuracy and variable selection. An empirical study is also carried out on the real Munich rent index data. This chapter is based on:

> **Zhang, B., Kneib, T., Bergherr, E.**. Bayesian-based Boosting for Quantifying Uncertainty in Structured Additive Regression. *working paper.*

Software

All of the analysis in this thesis was carried out on the statistical program `R` (R Core Team, 2019, 2020, 2021, 2022, depending on the time the respective research was done) in combination with related packages.

Chapter 2

Adaptive step-length selection in gradient boosting for Gaussian location and scale models

Generalized additive models for location, scale and shape (GAMLSS) (Rigby and Stasinopoulos, 2005) are distribution-based approaches, where all parameters of the assumed distribution for the response can be modelled as additive functions of the explanatory variables (Ripley, 2004; Stasinopoulos et al., 2017). Specifically, the GAMLSS framework allows the conditional distribution of the response variable to come from a wide variety of discrete, continuous and mixed discrete-continuous distributions, see Stasinopoulos and Rigby (2008). Unlike conventional generalized additive models (GAMs), GAMLSS not only model the location parameter, e.g. the mean for Gaussian distributions, but also further distribution parameters such as scale (variance) and shape (skewness and kurtosis) through the explanatory variables in linear, non-linear or smooth functional form.

The coefficients of GAMLSS are usually estimated based on penalized maximum likelihood method (Rigby and Stasinopoulos, 2005). However, this approach cannot deal with high-dimensional data, or more precisely, the case of more variables than observations (Bühlmann, 2006). As the selection of informative covariates is an important part of practical analysis, Mayr et al. (2012) combined the GAMLSS framework with componentwise gradient boosting (Bühlmann and Yu, 2003; Hofner et al., 2014; Hothorn et al., 2022) such that variable selection and estimation can be performed simultaneously. The original method cyclically updates the distribution parameters, i.e. all predictors will be updated sequentially in each boosting iteration (Hofner et al., 2016). Because the levels of complexity vary across the prediction functions, separate stopping values are required for each distribution parameter. Consequently, these stopping values have to be optimized jointly as they are not independent of each other. The commonly applied joint optimization methods like grid search are, however, computationally very demanding. For this reason, Thomas et al. (2018) proposed an alternative non-cyclical algorithm that updates only one distribution parameter (yielding the strongest improvement) in each boosting iteration. This way, only one global stopping value is needed and the resulting one-dimensional optimization procedure vastly reduces computing complexity for the boosting algorithm compared to the previous multi-dimensional one. The non-cyclical algorithm can be combined with stability selection (Meinshausen and Bühlmann, 2010; Hofner et al., 2015) to further reduce the selection of false positives (Hothorn et al., 2010).

In contrast to the cyclical approach, the non-cyclical algorithm avoids an equal number of updates for all distribution parameters as it is not useful to artificially enforce equal updates for parameters with a less complex structure than other parameters. However, it becomes even more important to fairly select the predictor to be updated in any given iteration. The current implementation of Thomas et al. (2018), however, uses fixed and equal step-lengths for all updates, regardless of the achieved loss reduction of different distribution parameters. In other words, different parameters affect the loss in different ways, and an update of the same size on all predictors hence results in different improvement with respect to loss reduction. As a consequence, a more useful update of one parameter could be rejected in favor of the other one just because the relevance in the loss function varies. As we demonstrate later, this leads to imbalanced updates that affect the fair selection and predictors with large number of boosting iterations still tend to be underfitted. This seems inconsistent, since one expects the underfitted predictor to be updated with a small number of iterations. As we show later, a large $\boldsymbol{\sigma}$ in a Gaussian distribution leads to a small negative gradient of $\boldsymbol{\mu}$ and consequently the improvement for $\boldsymbol{\mu}$ with fixed small step-length in each boosting iteration will also be small. This results in the algorithm needing a lot of updates for $\boldsymbol{\mu}$ until its empirical risk decreases to the level of $\boldsymbol{\sigma}$. However, the algorithm may stop long before the corresponding coefficients are well estimated.

We address this problem by proposing a variation of the non-cyclical boosting algorithm for GAMLSS, especially for Gaussian location and scale models, that adaptively and automatically optimizes step-lengths for all predictors in each boosting iteration. This ensures no parameter favored over the others by finding the factor that results in the overall best model improvement for each update and then bases the decision on which parameter to update on this comparison. While the adaptive approach does not enforce equal numbers of updates for all distribution parameters, it yields a fair selection of predictors to update and a natural balance in updates. For the very special Gaussian case, we also derive (semi-)analytical adaptive step-lengths that decrease the need for numerical optimization and discuss their properties. Our findings have implications beyond boosted Gaussian location and scale models for boosting other models with several predictors, e.g. the whole GAMLSS framework in general or for zero-inflated count models, and also give insights into the step-length choice for the simpler special case of (Gaussian) additive models.

The chapter is organized as follows: Section 2.1 introduces the boosted GAMLSS models including the cyclical and non-cyclical algorithms. Section 2.2 discusses how to apply the adaptive step-length on the non-cyclical boosted GAMLSS algorithm, and introduces the semi-analytical solutions of the adaptive step-length for the Gaussian location and scale models and discusses their properties. Section 2.3 evaluates the performance of the adaptive algorithms and the problem of fixed step-length in two simulations. Section 2.4 presents the application of the adaptive algorithms for two datasets: the malnutrition data, where the outcome variance is very large, and the riboflavin data, which has more variables than observations. Section 2.5 concludes with a summary and discussion. Further relevant materials and results are included in the appendix.

2.1 Boosted GAMLSS

In this section, we briefly introduce the GAMLSS models and the two cyclical and noncyclical boosting methods for estimation.

2.1.1 GAMLSS and componentwise gradient boosting

Conventional generalized additive models (GAM) assume a dependence only of the conditional mean μ of the response on the covariates. GAMLSS, however, also model other distribution parameters such as the scale σ, skewness ν and/or kurtosis τ with a set of statistical models.

The K distribution parameters $\boldsymbol{\theta}^T = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_K)$ of a density function $f(\boldsymbol{y}|\boldsymbol{\theta})$ are modelled by a set of up to K additive models. The model class assumes that the observations y_i for $i \in \{1, \cdots, n\}$ are conditionally independent given a set of explanatory variables. Let $\boldsymbol{y}^T = (y_1, y_2, \cdots, y_n)$ be a vector of the response variable and $\boldsymbol{X}$ be a $n \times J$ data matrix. In addition, we denote $\boldsymbol{X}_{i\cdot}$, $\boldsymbol{X}_{\cdot j}$ and X_{ij} as the i-th observation (vector of length J), j-variable (vector of length n) and the i-th observation of the j-th variable (a single value) respectively. Let $g_k(\cdot), k = 1, \cdots, K$ be known monotonic link functions that relate K distribution parameters to explanatory variables through additive models given by

$$g_k(\boldsymbol{\theta}_k) = \eta_{\boldsymbol{\theta}_k}(\boldsymbol{X}) = \beta_{0,\boldsymbol{\theta}_k}\mathbf{1}_n + \sum_{j=1}^{J} f_{j,\boldsymbol{\theta}_k}(\boldsymbol{X}_{\cdot j}|\beta_{j,\boldsymbol{\theta}_k}), \quad \text{for } k = 1, \ldots, K, \tag{2.1}$$

where $\boldsymbol{\theta}_k = (\theta_{k,1}, \cdots, \theta_{k,n})^T$ contains the n parameter values for the n observations and functions are applied elementwise if the argument is a vector, $\boldsymbol{\eta}_{\theta_k}$ is a vector of length n, $\mathbf{1}_n$ is a vector of ones and $\beta_{0,\boldsymbol{\theta}_k}$ is the model parameter specific intercept. Function $f_{j,\boldsymbol{\theta}_k}(\boldsymbol{X}_{\cdot j}|\beta_{j,\boldsymbol{\theta}_k})$ indicates the effects of the j-th explanatory variable $\boldsymbol{X}_{\cdot j}$ (vector of length n) for the model parameter $\boldsymbol{\theta}_k$, and $\beta_{j,\boldsymbol{\theta_k}}$ is the parameter of the additive predictor $f_{j,\boldsymbol{\theta}_k}(\cdot)$. Various types of effects (e.g., linear, smooth, random) for $f(\cdot)$ are allowed. If the location parameter ($\theta_1 = \mu$) is the only distribution parameter to be regressed ($K = 1$) and the response variable is from the exponential family, (2.1) reduces to the conventional GAM. In addition, f_j can depend on more than one variable (interaction), in which case $X_{\cdot j}$ would be e.g. a $n \times 2$ matrix, but for simplicity we ignore this case in the notation.

A penalized likelihood approach can be used to estimate the unknown quantities; for more details, see Rigby and Stasinopoulos (2005). This approach does not allow parameter estimation in the case of more explanatory variables than observations, and variable selection for high-dimensional data is not possible, which, however, can be well solved by using boosting. The theoretical foundations regarding numerical convergence and consistency of boosting with general loss functions have been studied by Zhang and Yu (2005). The work of Bühlmann and Yu (2003) on L_2 boosting with linear learners and Hastie et al. (2007) on the proof of the equivalence of the lasso and forward stagewise regression paved the way of componentwise gradient boosting (Hothorn et al., 2022), which emphasizes the importance of weak learners to reduce the tendency to overfit. To deal with the high-dimensional problems, Mayr et al. (2012) proposed a boosted GAMLSS algorithm, which estimates the predictors in GAMLSS with componentwise gradient boosting. As this method updates in general only one variable in each iteration, it can deal with data that has more variables than observations, and the important variables can be selected by controlling the stopping iterations.

To estimate the unknown predictor parameters $\beta_{j,\boldsymbol{\theta}_k}$, $j \in \{1, \cdots, J\}$ in equation (2.1), the componentwise gradient boosting algorithm minimizes the empirical risk R, which is also the loss ρ summed over all observations,

$$R = \sum_{i=1}^{n} \rho\left(y_i, \boldsymbol{\eta}(\boldsymbol{X}_{i\cdot})\right),$$

where the loss ρ measures the discrepancy between the response y_i and the predictor $\boldsymbol{\eta}(\boldsymbol{X}_{i\cdot})$. The predictor $\boldsymbol{\eta}(\boldsymbol{X}_{i\cdot}) = (\eta_{\boldsymbol{\theta}_1}(\boldsymbol{X}_{i\cdot}), \cdots, \eta_{\boldsymbol{\theta}_K}(\boldsymbol{X}_{i\cdot}))$ is a vector of length K. For the i-th observation $\boldsymbol{X}_{i\cdot}$, each predictor $\eta_{\boldsymbol{\theta}_k}(\boldsymbol{X}_{i\cdot})$ is a single value corresponding to the i-th entry in $\eta_{\boldsymbol{\theta_k}}$ in equation (2.1). The loss function ρ usually used in GAMLSS is the negative log-likelihood of the assumed distribution of $\boldsymbol{y}$ (Thomas et al., 2018; Friedman et al., 2000).

The main idea of gradient boosting is to fit simple regression base-learners $h_j(\cdot)$ to the pseudo-residuals vector $\boldsymbol{u}^T = (u_1, \cdots, u_n)$, which is defined as the negative partial derivatives of loss ρ, i.e.

$$\boldsymbol{u}_k^{[m]} = \left(-\frac{\partial}{\partial \eta_{\boldsymbol{\theta}_k}} \rho(y, \boldsymbol{\eta}) \Big|_{\boldsymbol{\eta} = \hat{\boldsymbol{\eta}}^{[m-1]}(\boldsymbol{X}_{i\cdot}), y = y_i} \right)_{i=1,\cdots,n},$$

where m denotes the current boosting iteration. In a componentwise gradient boosting iteration, each base-learner involves usually one explanatory variable (interactions are also allowed) and is fitted separately to $\boldsymbol{u}_k^{[m]}$,

$$\boldsymbol{u}_k^{[m]} \overset{\text{base-learner}}{\longrightarrow} \hat{h}_{j,\boldsymbol{\theta}_k}^{[m]}(\boldsymbol{X}_{\cdot j}) \quad \text{for} \quad j = 1, \cdots, J.$$

For linear base-learner, its correspondence to the model terms in (2.1) shall be

$$\hat{h}_{j,\boldsymbol{\theta}_k}(\boldsymbol{X}_{\cdot j}) = \boldsymbol{X}_{\cdot j}\hat{\boldsymbol{\beta}}_j,$$

where the estimated coefficients can be obtained by using the maximum likelihood or least square method. The best-fitting base-learner is selected based on the residual sum of squares, i.e.

$$j^* = \underset{j \in \{1,\cdots,J\}}{\arg\min} \sum_{i=1}^{n} \Big(u_{k,i} - \hat{h}_j(X_{ij}) \Big)^2,$$

thereby allowing for easy interpretability of the estimated model and also the use of hypothesis tests for single base-learners (Hepp et al., 2019). The additive predictor will be updated based on the best-fitting base-learner $\hat{h}_{j^*,\theta_{k^*}}(\boldsymbol{X}_{\cdot j^*})$ in terms of the best-performing sub-model $\eta_{\boldsymbol{\theta}_{k^*}}$,

$$\hat{\eta}_{\boldsymbol{\theta}_{k^*}}^{[m]}(\boldsymbol{X}) = \hat{\eta}_{\boldsymbol{\theta}_{k^*}}^{[m-1]}(\boldsymbol{X}) + \nu \hat{h}_{j^*,\boldsymbol{\theta}_{k^*}}(\boldsymbol{X}_{\cdot j^*}), \tag{2.2}$$

where ν denotes the step-length. In order to prevent overfitting, the step-length is usually set to a small value, in most cases 0.1. Equation (2.2) updates only the best-performing predictor $\hat{\eta}_{\boldsymbol{\theta}_{k^*}}^{[m]}$, all other predictors (i.e. for $k \neq k^*$) remain the same as in the previous boosting iteration. The best-performing sub-model $\boldsymbol{\theta}_{k^*}$ can be selected by comparing the empirical risk, i.e. which model parameter achieves the largest model improvement.

The main tuning parameter in this procedure, as in other boosting algorithms, is how many iterations should be performed before it stops, which is denoted as $m_{\theta_{\text{stop}}}$. As too large or small $m_{\theta_{\text{stop}}}$ leads to over-/underfitting model, cross-validation (Kohavi et al., 1995) is one of the most widely used methods to find the optimal $m_{\theta_{\text{stop}}}$.

2.1.2 Cyclical boosted GAMLSS

The boosted GAMLSS can deal with data that has more variables than observations, as the componentwise gradient boosting updates only one variable in each iteration. It leads to variable selection if some less important variables have never been selected as the best-performing variable and thus are not included in the final model for a given stopping iteration $m_{\theta_{\text{stop}}}$.

The original framework of boosted GAMLSS proposed by Mayr et al. (2012) is a cyclical approach, which means every predictor $\eta_{\boldsymbol{\theta}_k}, k \in \{1, \cdots, K\}$ is updated in a cyclical manner inside each boosting iteration. The iteration starts by updating the predictor for the location parameter and uses the predictors from the previous iteration for all other parameters. Then, the updated location model will be used for updating the scale model and so on. A schematic overview of the updating process in iteration $m+1$ for $K=4$ is

$$\begin{aligned}
\left(\hat{\boldsymbol{\mu}}^{[m]}, \hat{\boldsymbol{\sigma}}^{[m]}, \hat{\boldsymbol{\nu}}^{[m]}, \hat{\boldsymbol{\tau}}^{[m]}\right) &\overset{\text{update}}{\longrightarrow} \hat{\eta}_{\boldsymbol{\mu}}^{[m+1]} \rightarrow \hat{\boldsymbol{\mu}}^{[m+1]} \\
\left(\hat{\boldsymbol{\mu}}^{[m+1]}, \hat{\boldsymbol{\sigma}}^{[m]}, \hat{\boldsymbol{\nu}}^{[m]}, \hat{\boldsymbol{\tau}}^{[m]}\right) &\overset{\text{update}}{\longrightarrow} \hat{\eta}_{\boldsymbol{\sigma}}^{[m+1]} \rightarrow \hat{\boldsymbol{\sigma}}^{[m+1]} \\
\left(\hat{\boldsymbol{\mu}}^{[m+1]}, \hat{\boldsymbol{\sigma}}^{[m+1]}, \hat{\boldsymbol{\nu}}^{[m]}, \hat{\boldsymbol{\tau}}^{[m]}\right) &\overset{\text{update}}{\longrightarrow} \hat{\eta}_{\boldsymbol{\nu}}^{[m+1]} \rightarrow \hat{\boldsymbol{\nu}}^{[m+1]} \\
\left(\hat{\boldsymbol{\mu}}^{[m+1]}, \hat{\boldsymbol{\sigma}}^{[m+1]}, \hat{\boldsymbol{\nu}}^{[m+1]}, \hat{\boldsymbol{\tau}}^{[m]}\right) &\overset{\text{update}}{\longrightarrow} \hat{\eta}_{\boldsymbol{\tau}}^{[m+1]} \rightarrow \hat{\boldsymbol{\tau}}^{[m+1]}.
\end{aligned}$$

However, not all of the distribution parameters have the same complexity, i.e. the stopping iterations $m_{\theta_{\text{stop}}}$ should be set separately for different parameters, or jointly optimized, for example by grid search. Since grid search scales exponentially with the number of distribution parameters, such optimization can be very slow.

2.1.3 Non-cyclical boosted GAMLSS

In order to deal with the issues of a cyclical approach, Thomas et al. (2018) proposed a *non-cyclical* variation, that updates only one distribution parameter instead of successively updating all parameters in each boosting iteration by comparing the model improvement (negative log-likelihood) of each model parameter, see algorithm 1 (especially step 11). Consequently, instead of specifying separate stopping iterations $m_{\boldsymbol{\theta}\text{stop}}$ for different parameters and tuning them with the computationally demanding grid search, only one overall stopping iteration, denoted as m_{stop}, needs to be tuned with e.g. the line search (Friedman, 2001; Brent, 2013). The tuning problem thus reduces from a multi-dimensional to a one-dimensional problem, which vastly reduces the computing time.

Algorithm 1 has a nested structure, with the outer loop executing the boosting iterations and the inner loops addressing the different distribution parameters. The best-fitting base-learner and their contribution to the model improvement for every parameter is selected in the inner loop and compared in the outer loop (step 11). Therefore, only the best-performing base-learner is updated in a single iteration by adding $\nu\hat{h}(X_{\cdot j^*})$ to the predictor of the corresponding parameter θ_{k^*}. Over the course of the iterations, the boosting algorithm steadily increases the model in small steps and the final estimates for different base-learners are simply the sum of all their updates they may have received.

The cyclical approach led to an inherent but somewhat artificial balance between the distribution parameters, as predictors for all distribution parameters are updated in each iteration. Different final stopping values $m_{\boldsymbol{\theta}\text{stop}}$ for different distribution parameters - chosen by tuning methods such as cross-validation - allow stopping updates at different times for distribution parameters of different complexity to avoid overfitting. In the non-cyclical algorithm, especially when m_{stop} is not large enough, there is the danger of an imbalance between predictors. If the selection between predictors to update is not fair, this could lead to iterations primarily updating some of the predictors and underfitting others. We will provide a detailed example for the Gaussian distribution with large $\boldsymbol{\sigma}$ in Section 2.3.2.

A related challenge is to choose an appropriate step-length $\nu_{\boldsymbol{\theta}_k}^{[m]}$ for both the cyclical and non-cyclical approaches. Tuning the parameters when boosting GAMLSS models relies mainly on the number of boosting iterations (m_{stop}), with the step-length ν usually set to a small value such as 0.1. Bühlmann and Hothorn (2007) argued that using a small step-length like 0.1 (potentially resulting in a larger number of iterations m_{stop}) had a similar computing speed as using an adaptive step-length performed by doing a line search, but meant an easier tuning task for one parameter (m_{stop}) instead of two (m_{stop} and ν). However, this results referred to models with a single predictor. A fixed step-length can lead to an imbalance in the case of several predictors that may live on quite different scales. For example, 0.1 may be too small for μ but large for

Algorithm 1 Non-cyclical componentwise gradient boosting in multiple dimensions - Basic algorithm

```
Initialize additive predictors $\hat{\boldsymbol{\eta}}^{[0]} = \left(\hat{\eta}^{[0]}_{\boldsymbol{\theta}_1}, \cdots, \hat{\eta}^{[0]}_{\boldsymbol{\theta}_K}\right)$ with offsets.
For each distribution parameter $\boldsymbol{\theta}_k, k = 1, \cdots, K$, specify a set of base-learners, i.e. for parameter $\boldsymbol{\theta}_k$ define $h_{1,\boldsymbol{\theta}_k}(\cdot), \cdots, h_{J_k,\boldsymbol{\theta}_k}(\cdot)$ where $J_k$ is the cardinality of the set of base-learners specified for $\boldsymbol{\theta}_k$.
for $m = 1$ to $m_{\text{stop}}$ do
    for $k = 1$ to $K$ do
        Compute negative partial derivatives $-\frac{\partial}{\partial \eta_{\boldsymbol{\theta}_k}} \rho(y, \boldsymbol{\eta})$ and plug in the current estimates $\hat{\boldsymbol{\eta}}^{[m-1]}(\cdot)$:
            $$\boldsymbol{u}_k^{[m]} = -\frac{\partial}{\partial \eta_{\boldsymbol{\theta}_k}} \rho(y, \boldsymbol{\eta}),$$
        where $\boldsymbol{\eta} = \hat{\boldsymbol{\eta}}^{[m-1]}(\boldsymbol{X}_{i\cdot})$ and $y = y_i$ for $i = 1, \cdots, n$.
        Fit (e.g. with the least square method) the negative gradient vector $\boldsymbol{u}_k^{[m]}$ separately to every base-learner:
            $$\boldsymbol{u}_k^{[m]} \overset{\text{base-learner}}{\longrightarrow} \hat{h}_{j,\boldsymbol{\theta}_k}(\boldsymbol{X}_{\cdot j}) \quad \text{for } j = 1, \cdots, J_k.$$
        Select the best-fitting base-learner $\hat{h}_{j^*,\boldsymbol{\theta}_k}(\boldsymbol{X}_{\cdot j^*})$ by inner loss, i.e. the residual sum of squares of the base-learner fit w.r.t. $\boldsymbol{u}_k^{[m]} = \left(u_{k,1}^{[m]}, \cdots, u_{k,n}^{[m]}\right)^T$:
            $$j^* = \underset{j \in \{1,\cdots,J_k\}}{\arg\min} \sum_{i=1}^{n} \left(u_{k,i}^{[m]} - \hat{h}_{j,\boldsymbol{\theta}_k}(X_{ij})\right)^2,$$
        where we dropped the dependence of $j^*$ on $k$ in the notation for simplicity.
        Set the step-length to a fixed value $\nu_0$, usually $\nu_0 = 0.1$:
            $$\nu_{\boldsymbol{\theta}_k}^{[m]} = \nu_0$$
        Compute the possible improvement of this update regarding the outer loss
            $$\Delta\rho_k = \sum_{i=1}^{n} \rho\left(y_i, \hat{\eta}_{\boldsymbol{\theta}_k}^{[m-1]}(\boldsymbol{X}_{i\cdot}) + \nu_{\boldsymbol{\theta}_k}^{[m]} \cdot \hat{h}_{j^*,\boldsymbol{\theta}_k}(X_{ij^*})\right).$$
    end for
    Update, depending on the value of the loss reduction, only the overall best-fitting base-learner $k^* = \arg\min_{k \in \{1,\cdots,K\}} \Delta\rho_k$:
        $$\hat{\eta}_{\boldsymbol{\theta}_{k^*}}^{[m]}(\boldsymbol{X}) = \hat{\eta}_{\boldsymbol{\theta}_{k^*}}^{[m-1]}(\boldsymbol{X}) + \nu_{\boldsymbol{\theta}_k}^{[m]} \cdot \hat{h}_{j^*,\boldsymbol{\theta}_{k^*}}(\boldsymbol{X}_{\cdot j^*}).$$
    Set $\hat{\boldsymbol{\eta}}_{\boldsymbol{\theta}_k}^{[m]} := \hat{\boldsymbol{\eta}}_{\boldsymbol{\theta}_k}^{[m-1]}$ for all $k \neq k^*$.
end for
```

σ. We will discuss such cases analytically and with empirical evidence in the later sections. Moreover, varying the step-lengths for different sub-models directly influences the choice of best-performing sub-model in the non-cyclical boosting algorithm, thus choosing a subjective step-length is not appropriate. In the following, we denote a fixed predefined step-length such as 0.1 as the *fixed step-length* (FSL) approach.

To overcome the problems stated above, we suggest to use *adaptive step-lengths* (ASL) while boosting. In particular, we propose to optimize the step-length for each predictor in each iteration to obtain a fair comparison between the predictors. While the adaptive step-length has been used before, the proposal to use different ASLs for different predictors is new and we will see that this leads to balanced updates of the different predictors.

2.2 Adaptive Step-Length

In this section, we first introduce the general idea of the implementation of adaptive step-lengths for different predictors to GAMLSS. For the important special case of a Gaussian location and scale models with two model parameters (μ and σ), we will derive and discuss their analytical adaptive step-lengths and properties, which also serves as an important illustration of the relevant issues more generally.

2.2.1 Boosted GAMLSS with adaptive step-length

Unlike the step-length in equation (2.2) and algorithm 1, step 11, the adaptive step-length may also vary in different boosting iterations according to the loss reduction.

The adaptive step-length can be derived by solving the optimization problem

$$\nu^{*[m]}_{j^*,\boldsymbol{\theta}_k} = \arg\min_{\nu} \sum_{i=1}^{n} \rho\left(y_i, \hat{\eta}^{[m-1]}_{\boldsymbol{\theta}_k}(\boldsymbol{X}_{i\cdot}) + \nu \cdot \hat{h}_{j^*,\boldsymbol{\theta}_k}(X_{ij^*})\right), \tag{2.3}$$

note that $\nu^{*[m]}_{j^*,\boldsymbol{\theta}_k}$ is the *optimal step-length* of the model parameter θ_k dependent on j^* in iteration m. The optimal step-length is a value that leads to the largest decrease possible of the empirical risk and usually leads to overfitting of the corresponding variable if no shrinkage is used (Hepp et al., 2016). Therefore the actual adaptive step-length (ASL) we apply in the boosting algorithm is the product of two parts, the

Algorithm 2 Non-cyclical componentwise gradient boosting with adaptive step-length - Extension of basic algorithm 1

```
   ··· Steps 1-7 equal to algorithm 1 ···, in addition, choose shrinkage parameter $\lambda$.
8: Find the optimal step-length $\nu_{\boldsymbol{\theta}_k}^{[m]}$ by optimizing the outer loss:
       $\nu_{j^*,\boldsymbol{\theta}_k}^{*[m]} = \arg\min_{\nu} \sum_{i=1}^{n} \rho\left(y_i, \hat{\eta}_{\boldsymbol{\theta}_k}^{[m-1]}(\boldsymbol{X}_{i\cdot}) + \nu \cdot \hat{h}_{j^*,\boldsymbol{\theta}_k}(X_{ij^*})\right),$
   and set adaptive step-length $\nu_{j^*,\boldsymbol{\theta}_k}^{[m]}$ as the optimal value with shrinkage $\lambda$:
       $\nu_{j^*,\boldsymbol{\theta}_k}^{[m]} = \lambda \cdot \nu_{j^*,\boldsymbol{\theta}_k}^{*[m]}.$
   ··· Steps 9-13 equal to those in algorithm 1 ···
```

shrinkage parameter λ and the optimal step-length $\nu_{j^*,\boldsymbol{\theta}_k}^{*[m]}$, i.e.

$$\nu_{j^*,\boldsymbol{\theta}_k}^{[m]} = \lambda \cdot \nu_{j^*,\boldsymbol{\theta}_k}^{*[m]}.$$

In this chapter, we take $\lambda = 0.1$, thus 10% of the optimal step-length. By comparison, the fixed step-length $\nu = 0.1$ would correspond to a combination of a shrinkage parameter $\lambda = 0.1$ with the "optimal" step-length ν^* set to one.

The non-cyclical algorithm with ASL can be improved by replacing the fixed step-length in step 8 of algorithm 1 with the adaptive one. We formulate this change in algorithm 2.

As the parameters in GAMLSS may have quite different scales, updates with fixed step-length can lead to an imbalance between sub-models, especially when m_{stop} is not large enough. When using FSL, a single update for predictor $\eta_{\boldsymbol{\theta}_1}$ may achieve the same amount of global loss reduction than several updates of another predictor $\eta_{\boldsymbol{\theta}_2}$ even if the actually possible contribution of the competing base-learners is similar, because for different scales the loss reduction of $\eta_{\boldsymbol{\theta}_2}$ in these iterations are always smaller than that of $\eta_{\boldsymbol{\theta}_1}$. However, such unfair selections can be avoided by using ASL, because the model improvement depends on the largest decrease possible of each predictor, i.e. the potential reduction in the empirical risks of all predictors are on the same level and their comparison therefore is fair. Fair selection does not enforce an equal number of updates as in the cyclical approach. The ASL approach can lead to imbalanced updates of predictors, but such imbalance actually reveals the intrinsically different complexities of each sub-model.

The main contribution of this chapter is the proposal to use ASLs for each predictor in GAMLSS. This idea can also be applied to other complex models (e.g. zero-inflated count models) with several predictors for the different parameters, because these models meet the same problem, i.e. the scale of these parameters might differ considerably. If a boosting algorithm is preferred for estimation of such a model, we provide a new solution to address these kinds of problems, i.e. separate adaptive step-lengths for each distribution parameter.

2.2.2 Gaussian location and scale models

In general, the adaptive step-length ν can be found by optimizing procedures such as a line search. However, such methods do not help to reveal the properties of adaptive step-lengths and its relationship with model parameters. Moreover, a line search method searches for the optimal value from a predefined search interval, which can be difficult to find out since too narrow intervals might not include the optimal value and too large intervals increase the searching time. The direct computation from an analytical expression is faster than a search. By investigating the important special case of a Gaussian distribution with two parameters, we will learn a lot about the adaptive step-length for the general case. Nevertheless, we must underline that for many cases an explicit closed form for the adaptive step-length may not exist and line search still plays an irreplaceable role. We perform the following study of the analytical solutions for the Gaussian special case out of the wish of finding its inner relationship with the model parameters, in order to better understand the limitation of fixed step-length and how adaptive values improve the learning process.

Consider the data points $(y_i, \boldsymbol{x}_{i\cdot}), i \in \{1, \cdots, n\}$, where $\boldsymbol{x}$ is a $n \times J$ matrix. Assuming that the true data generating mechanism is a Gaussian model

$$\begin{aligned} & y_i \sim \mathrm{N}(\mu_i, \sigma_i) \\ & \mu_i = \eta_{\boldsymbol{\mu}}(\boldsymbol{x}_{i\cdot}) \\ & \sigma_i = \exp\left(\eta_{\boldsymbol{\sigma}}(\boldsymbol{x}_{i\cdot})\right). \end{aligned}$$

As we talk about the observed data, we replace $\eta_{\boldsymbol{\theta}_k}$, where $k = 1, 2$ for Gaussian distribution, with $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, and replace $\boldsymbol{X}$ with $\boldsymbol{x}$. The identity and exponential functions for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are thus the corresponding inverse link. Taking the negative log-likelihood as the loss function, its negative partial derivatives $\boldsymbol{u}_{\boldsymbol{\mu}}$ and $\boldsymbol{u}_{\boldsymbol{\sigma}}$ in iteration m for both parameters can then be modelled with the base-learners $\hat{h}_{j,\boldsymbol{\mu}}^{[m]}$ and $\hat{h}_{j,\boldsymbol{\sigma}}^{[m]}$. The optimization process can then be divided into two parts: one is the ASL for the location parameter $\boldsymbol{\mu}$, and the other is for the scale parameter $\boldsymbol{\sigma}$. As the ASL shrinks the optimal value, we consider only the optimal step-lengths for both parameters.

Optimal step-length for $\boldsymbol{\mu}$

The analytical optimal step-length for $\boldsymbol{\mu}$ in iteration m is obtained by minimizing the empirical risk

$$
\begin{aligned}
\nu_{j^*,\boldsymbol{\mu}}^{*[m]} &= \arg\min_{\nu} \sum_{i=1}^{n} \rho\left(y_i, \{\hat{\eta}_{\boldsymbol{\mu}}^{[m]}(\boldsymbol{x}_{i\cdot}), \hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\}\right) \\
&= \arg\min_{\nu} \sum_{i=1}^{n} \frac{\left(y_i - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) - \nu \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{2\hat{\sigma}_i^{2[m-1]}},
\end{aligned}
\tag{2.4}
$$

where the expression $\hat{\sigma}_i^{2[m-1]}$ represents the square of the standard deviation in the previous iteration, i.e. $\hat{\sigma}_i^{2[m-1]} = (\hat{\sigma}_i^{[m-1]})^2$. The optimal value of $\nu_{j^*,\boldsymbol{\mu}}^{*[m]}$ is obtained by letting the derivative of the equation equal zero, so we get the analytical ASL for $\boldsymbol{\mu}$ (for more derivation details, see also appendix A.1.1):

$$
\nu_{j^*,\boldsymbol{\mu}}^{*[m]} = \frac{\sum_{i=1}^{n} \left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\sum_{i=1}^{n} \frac{\left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\hat{\sigma}_i^{2[m-1]}}}. \tag{2.5}
$$

It is obvious, that $\nu_{j^*,\boldsymbol{\mu}}^{*[m]}$ is not an independent parameter in GAMLSS but depends on the base-learner $\hat{h}_{\boldsymbol{\mu}}^{[m]}(x_{ij^*})$ with respect to the best performing variable $\boldsymbol{x}_{\cdot j^*}$ and the estimated variance in the previous iteration $\hat{\sigma}_i^{2[m-1]}$.

In the special case of a Gaussian additive model, the scale parameter σ is assumed to be constant, i.e. $\hat{\sigma}_i^{[m-1]} = \hat{\sigma}^{[m-1]}$ for all $i \in \{1, \cdots, n\}$. We then obtain

$$
\nu_{j^*,\boldsymbol{\mu}}^{*[m]} = \frac{\sum_{i=1}^{n} \left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\frac{1}{\hat{\sigma}^{2[m-1]}} \sum_{i=1}^{n} \left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2} = \hat{\sigma}^{2[m-1]}. \tag{2.6}
$$

This gives us an interesting property of the optimal step-length, i.e. the analytical optimal step-length for μ in the Gaussian distribution is actually the variance (as computed in the previous boosting iteration). This property enables this adaptive step-length to be not only applicable for the special GAMLSS case, but also for the boosting of additive models with normal responses. Therefore, in the case of Gaussian additive models, we can use $\nu_{j^*,\boldsymbol{\mu}}^{[m]} = \lambda\hat{\sigma}^{2[m-1]}$ as the step-length, which has a stronger theoretical foundation, instead of the common choice 0.1.

Back to the general GAMLSS case, we can further investigate the behavior of the step-length by considering the limiting case of $m \to \infty$. For large m, all base-learner fits $\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})$ converge to zero or are similarly small. If we consequently approximate all $\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})$ by some small constant h, this gives an approximation of the analytical optimal step-length of

$$\nu_{j^*,\boldsymbol{\mu}}^{*[m]} \approx \frac{\sum_{i=1}^{n} h^2}{\sum_{i=1}^{n} \frac{h^2}{\hat{\sigma}_i^{2[m-1]}}} = \frac{nh^2}{h^2 \sum_{i=1}^{n} \frac{1}{\hat{\sigma}_i^{2[m-1]}}} = \frac{n}{\sum_{i=1}^{n} \frac{1}{\hat{\sigma}_i^{2[m-1]}}}, \tag{2.7}$$

which is the harmonic mean of the estimated variances $\hat{\sigma}_i^{2[m-1]}$ in the previous iteration. While this expression requires m to be large, which may not be reached if m_{stop} is of moderate size to prevent overfitting, the expression still gives an indication of the strong dependence of the optimal step-length on the variances $\hat{\sigma}_i^{2[m-1]}$, which generalizes the optimal value of the additive model in (2.6).

Optimal step-length for $\boldsymbol{\sigma}$

The optimal step-length for the scale parameter $\boldsymbol{\sigma}$ can be obtained analogously by minimizing the empirical risk, now with respect to $\nu_{j^*,\boldsymbol{\sigma}}^{*[m]}$. We obtain

$$\begin{aligned}
\nu_{j^*,\boldsymbol{\sigma}}^{*[m]} &= \arg\min_{\nu} \sum_{i=1}^{n} \rho\left(y_i, \{\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}), \hat{\eta}_{\boldsymbol{\sigma}}^{[m]}(\boldsymbol{x}_{i\cdot})\}\right) \\
&= \arg\min_{\nu} \sum_{i=1}^{n} \left(\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) + \nu\hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*})\right) + \\
&\quad + \sum_{i=1}^{n} \frac{\left(y_i - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\right)^2}{2\exp\left(2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) + 2\nu\hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*})\right)}.
\end{aligned} \tag{2.8}$$

After checking the positivity of the second-order derivative of the expression in equation (2.8), the optimal value can be obtained by setting the first-order derivative equal to zero:

$$\sum_{i=1}^{n} \hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*}) - \sum_{i=1}^{n} \frac{\left(\hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*}) + \epsilon_{i,\boldsymbol{\sigma}} + 1\right) \hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*})}{\exp\left(2\nu_{j^*,\boldsymbol{\sigma}}^{*[m]} \hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*})\right)} \overset{!}{=} 0, \tag{2.9}$$

where $\epsilon_{i,\boldsymbol{\sigma}}$ denotes the residuals when regressing the negative partial derivatives $\boldsymbol{u}_{\boldsymbol{\sigma},i}^{[m]}$ on the base-learner $\hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*})$, i.e. $u_{\boldsymbol{\sigma},i} = \hat{h}_{\boldsymbol{\sigma}}^{[m]}(\boldsymbol{x}_{i\cdot}) + \epsilon_{i,\boldsymbol{\sigma}}$. Unfortunately, equation (2.9) cannot be further simplified, which means that there is no analytical ASL for the scale parameter $\boldsymbol{\sigma}$ in the Gaussian distribution. Hence, the optimal ASL must be found by performing a conventional line search. For more details, see also Appendix A.1.2.

Even without an analytical solution, we can still use (2.9) to further study the behavior of the ASL. Analogous to the derivation of (2.7), $\hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*})$ converges to zero for $m \to \infty$. If we approximate with a (small) constant $\hat{h}_{\boldsymbol{\sigma}}^{[m]}(x_{ij^*}) \approx h, \forall i \in \{1, \cdots, n\}$. Then (2.9) simplifies to

$$\begin{aligned}
\sum_{i=1}^{n} h - \sum_{i=1}^{n} \frac{(h + \epsilon_{i,\boldsymbol{\sigma}} + 1)h}{\exp\left(2\nu_{j^*,\boldsymbol{\sigma}}^{*[m]} h\right)} &= 0 \\
\Leftrightarrow \nu_{j^*,\boldsymbol{\sigma}}^{*[m]} &= \frac{1}{2h} \log\left(h + 1 + \frac{1}{n}\sum_{i=1}^{n} \epsilon_{i,\boldsymbol{\sigma}}\right) \\
\Leftrightarrow \nu_{j^*,\boldsymbol{\sigma}}^{*[m]} &= \frac{1}{2h} \log(h+1),
\end{aligned} \tag{2.10}$$

where $\frac{1}{n}\sum_{i=1}^{n} \epsilon_{i,\boldsymbol{\sigma}} = 0$ in the regression model. Equation (2.10) can be further simplified by approximating the logarithm function with a Taylor series at $h = 0$, thus

$$\begin{aligned}
\nu_{j^*,\boldsymbol{\sigma}}^{*[m]} &= \frac{1}{2h}\left(h - \frac{h^2}{2} + O(h^3)\right) \\
&= \frac{1}{2} - \frac{h}{4} + O(h^2).
\end{aligned}$$

As $h \to 0$ for $m \to \infty$, the limit of this approximate optimal step-length for σ is

$$\lim_{m\to\infty} \nu_{j^*,\boldsymbol{\sigma}}^{*[m]} = \lim_{h\to 0} \frac{1}{2} - \frac{h}{4} = \frac{1}{2}. \tag{2.11}$$

Thus, the ASL for $\boldsymbol{\sigma}$ approaches approximately 0.05 if we take the shrinkage parameter

$\lambda = 0.1$ and iterations run for a longer time (and the boosting algorithm is not stopped too early to prevent overfitting for this trend to show).

2.2.3 (Semi-)Analytical adaptive step-length

Knowing the properties of the analytical ASL in boosting GAMLSS for the Gaussian distribution, we can replace the line search with the analytical solution for the location parameter $\boldsymbol{\mu}$. If we keep the line search for the scale parameter $\boldsymbol{\sigma}$, we call this the *Semi-Analytical Adaptive Step-Length (SAASL)*. Moreover, we are interested in the performance of combining the analytical ASL for $\boldsymbol{\mu}$ with the approximate value $0.05 = \lambda \cdot \frac{1}{2}$ (with $\lambda = 0.1$) for the ASL for $\boldsymbol{\sigma}$, which is motivated by the limiting considerations discussed above and has a better theoretical foundation than selecting an arbitrary small value in the common FSL. We call this step-length setup *SAASL05*. In either of these cases, it is straightforward and computationally efficient to obtain the (approximate) optimal value(s) and both alternatives are faster than performing two line searches.

The semi-analytical solution avoids the need for selecting a search interval for the line search, at least for the ASL for $\boldsymbol{\mu}$ in the case of SAASL and for both parameters for SAASL05. This is an advantage, since too large search intervals will cause additional computing time, but too small intervals may miss the optimal ASL value and again lead to an imbalance of updates between the parameters. Also note that the value 0.5 gives an indication for a reasonable range for the search interval for $\nu_{j^*,\boldsymbol{\sigma}}^{*[m]}$ if a line search is conducted after all.

The boosting GAMLSS algorithm with ASL for the Gaussian distribution is shown in algorithm 3.

For a chosen shrinkage parameter of $\lambda = 0.1$, the $\nu_{\boldsymbol{\sigma}}$ in SAASL05 would be 0.05, which is a smaller or “less aggressive” value than 0.1 in FSL, leading to a somewhat larger number of boosting iterations but a smaller risk of overfitting, and to a better balance with the ASL for $\boldsymbol{\mu}$.

Algorithm 3 Non-cyclical componentwise gradient boosting for the Gaussian location and scale models with different step-lengths - Extension of basic algorithm 1

```
   ··· Steps 1-7 equal to algorithm 1 ···, in addition, choose shrinkage parameter $\lambda$.
8: Set or find the step-length $\nu_{j^*,\boldsymbol{\theta}_k}^{[m]}$ for $\boldsymbol{\theta}_k \in \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ by one of the followings:
       • Adaptive step-length (ASL):
             $\nu_{j^*,\boldsymbol{\theta}_k}^{*[m]} = \arg\min_{\nu} \sum_{i=1}^{n} \rho\left(y_i, \hat{\eta}_{\boldsymbol{\theta}_k}^{[m-1]}(\boldsymbol{x}_{i\cdot}) + \nu \cdot \hat{h}_{j^*,\boldsymbol{\theta}_k}(x_{ij^*})\right);$
       • Semi-analytical adaptive step-length (SAASL):
         if $\boldsymbol{\theta}_k = \boldsymbol{\mu}$,
             $\nu_{j^*,\boldsymbol{\mu}}^{*[m]} = \frac{\sum_{i=1}^{n}\left(\hat{h}_{j^*,\boldsymbol{\mu}}(x_{ij^*})\right)^2}{\sum_{i=1}^{n}\frac{\left(\hat{h}_{j^*,\boldsymbol{\mu}}(x_{ij^*})\right)^2}{\hat{\sigma}_i^{2[m-1]}}},$
         if $\boldsymbol{\theta}_k = \boldsymbol{\sigma}$, same as for ASL.
       • Semi-analytical adaptive step-length (SAASL05):
         if $\boldsymbol{\theta}_k = \boldsymbol{\mu}$, same as for SAASL,
         if $\boldsymbol{\theta}_k = \boldsymbol{\sigma}$, $\nu_{j^*,\boldsymbol{\theta}_k}^{*[m]} = 0.5$.
   and set adaptive step-length $\nu_{j^*,\boldsymbol{\theta}_k}^{[m]}$ as the optimal value with shrinkage $\lambda$:
             $\nu_{j^*,\boldsymbol{\theta}_k}^{[m]} = \lambda \cdot \nu_{j^*,\boldsymbol{\theta}_k}^{*[m]}.$
   ··· Steps 9-13 equal to those in algorithm 1 ···
```

2.3 Simulation Study

In the following, two simulations are shown to demonstrate the performance of the adaptive algorithms. The first one compares the estimation accuracy between different non-cyclical boosted GAMLSS algorithms with FSL or ASL in a Gaussian regression model for location and scale. The second one underlines the problem of FSL and the performance of adaptive approaches if the variance in this setting is large.

2.3.1 Gaussian Location and Scale Model

The simulation study in Thomas et al. (2018) showed that their FSL non-cyclical approach outperforms the classical cyclical approach. We use the same setup to show that the ASL approach performs at least as good as the FSL non-cyclical approach (and hence also outperforms the classical cyclical approach). At the end of this subsection we will show that the reason for the good performance of FSL is due to the chosen simulated data structure. The setup is the following: the response y_i is drawn from $\mathrm{N}(\mu_i, \sigma_i)$ for $n = 500$ observations, with 6 informative covariates $x_{ij}, j \in \{1, \cdots, 6\}$ drawn independently from $\mathrm{U}(-1, 1)$. The predictors of both distribution parameters are:

$$\begin{aligned}\eta_{\boldsymbol{\mu}}(\boldsymbol{x}_{i\cdot}) &= \mu_i = x_{i1} + 2x_{i2} + 0.5x_{i3} - x_{i4}\\ \eta_{\boldsymbol{\sigma}}(\boldsymbol{x}_{i\cdot}) &= \log(\sigma_i) = 0.5x_{i3} + 0.25x_{i4} - 0.25x_{i5} - 0.5x_{i6},\end{aligned}$$

where x_3 and x_4 are shared between both μ and σ.

Moreover, $p_{\text{n-inf}} = 0, 50, 250$ or 500 non-informative variables sampled from $\mathrm{U}(-1, 1)$ are also added to the model. We conduct $B = 100$ simulation runs.

The estimated coefficients of $\eta_{\boldsymbol{\mu}}$ and $\eta_{\boldsymbol{\sigma}}$, whose values are taken at stopping iterations tuned by 10-fold CV with the maximum number of boosting iterations set to 1000, are shown in appendix figures A.1 and A.2. Overall, estimated coefficients are similar between all four methods, with the shrinkage bias of boosting only becoming apparent with an increasing number of noise variables.

Figure 2.1 shows the comparison of the mean squared error (MSE) among non-cyclical boosted algorithms for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, where the MSEs are defined on the predictor level as $\mathrm{MSE}_{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^{n}(\mu_i - \eta_{\boldsymbol{\mu}}(\boldsymbol{x}_{i\cdot}))^2$ and $\mathrm{MSE}_{\boldsymbol{\sigma}} = \frac{1}{n}\sum_{i=1}^{n}(\log(\sigma_i) - \eta_{\boldsymbol{\sigma}}(\boldsymbol{x}_{i\cdot}))^2$, respectively. In general, all methods have a similar MSE, with the MSE of FSL increasing more strongly with the number of non-informative variables $p_{\text{n-inf}}$ and ASL methods hence slightly outperform FSL in the variance predictor for a high number of non-informative variables. ASL and SAASL show identical results, as they should if the line search is correctly conducted, with results returned by SAASL05 very similar.

Computing the negative log-likelihood in sample of the model fits reveals a slight advantage for FSL (see appendix figure A.3). However, this can be linked to the fact that FSL selects more false positive variables on average than the adaptive approaches and thus shows a relatively stronger tendency to overfit the training data (figure 2.2).

Figure 2.2 illustrates the false positives of each methods for each parameter. For $\boldsymbol{\sigma}$, even if $p_{\text{n-info}}$ is small, the false positive rates of the adaptive approaches are notably

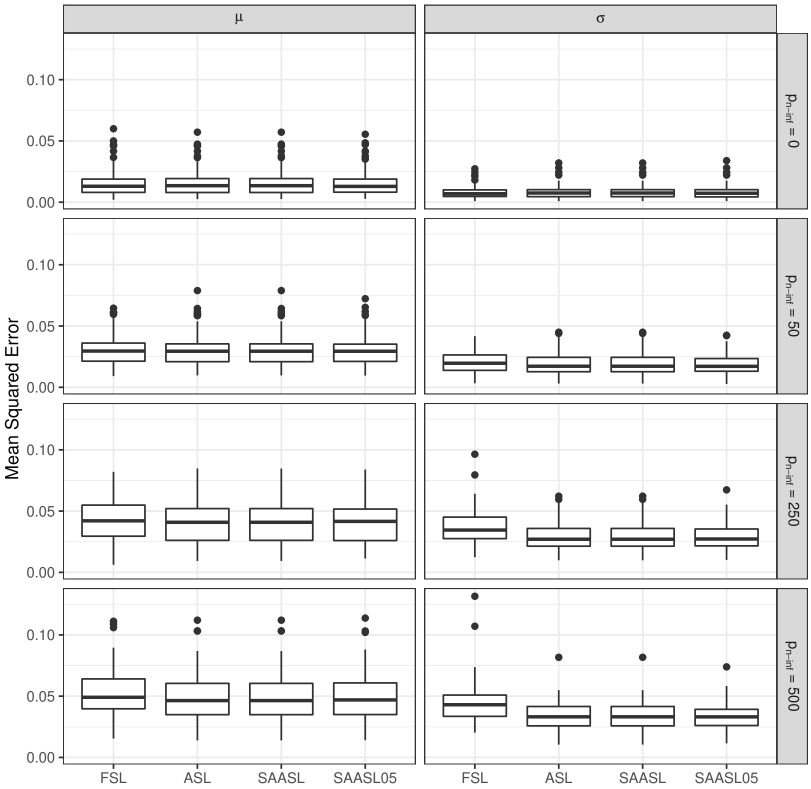


Figure 2.1 Comparison between mean squared error for FSL and the three ASL methods. The left column comprises the MSE for $\eta_{\boldsymbol{\mu}}$, the right column for $\eta_{\boldsymbol{\sigma}}$. The different numbers of non-informative variables are represented row-wise.

smaller than those of FSL. As discussed above, $\nu^{[m]}_{j^*,\boldsymbol{\sigma}} \approx 0.05$ for large m in the adaptive approach is smaller than $\nu_{\boldsymbol{\sigma}} = 0.1$ for FSL. An update with a smaller, conservative step-length can apparently help to avoid overfitting and the adaptive step-length here seems to strike the balance between learning speed and the number of false positives. While it would also be possible to lower the step-length for FSL to reduce the number of non-informative variables included in the final model, this would increase the number of boosting iterations and the computing time, and it would not address the imbalance between updates for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. The optimal choice of the step-length is also difficult without further tuning or an automatic selection as in ASL.

With respect to the neglecting of actually informative variables, i.e. false negatives, all four methods are able to find and select all variables for $\boldsymbol{\mu}$ in all of the simulation runs. Regarding $\boldsymbol{\sigma}$, the risk of false negatives slightly increases with the number of noise variables in the setting. However, even in the case of 500 noise-variables, only a single false negative is observed in between 3% and 6% of the runs, independently of the algorithm in question.

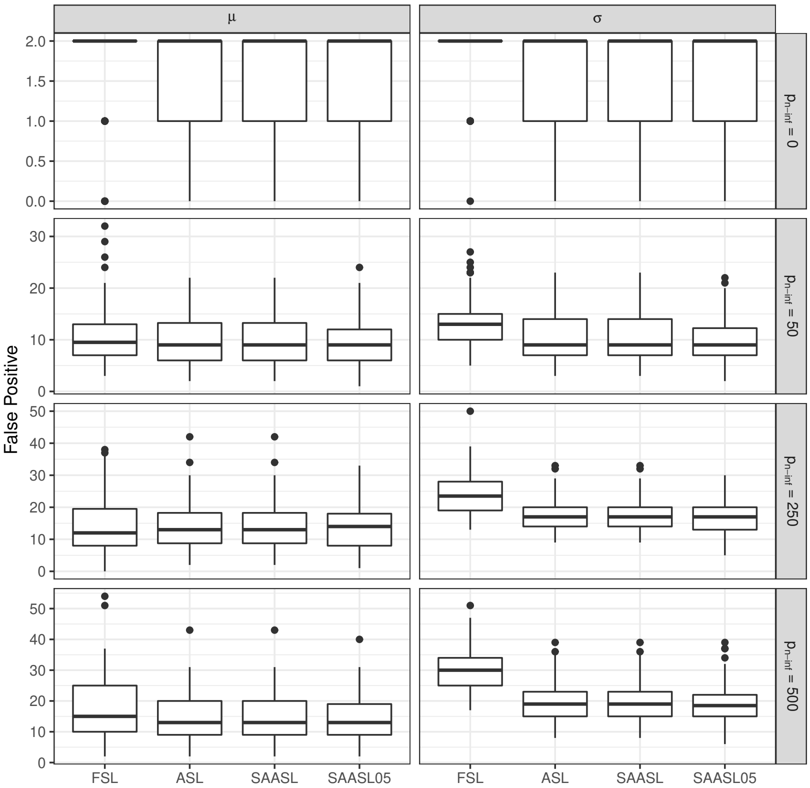


Figure 2.2 Comparison between false positives for FSL and the three ASL methods. The left column comprises the false positives for $\boldsymbol{\mu}$, and the right column for $\boldsymbol{\sigma}$. The different numbers of non-informative variables settings are represented row-wise.

To some extent, the low false negative rate can be expected considering the somewhat greedy nature of boosting algorithms. For this reason, performance in terms of false-positive selections is arguably the more important aspect and speaks to the adaptive updates.

In figure 2.3 we show an example of the comparison between the optimal step-lengths in this case. As can be seen, the step-lengths for $\boldsymbol{\sigma}$ (depicted in grey) converge to 0.5 as shown in section 2.2.2. The second fact that becomes obvious when looking at the figure is that the optimal step-lengths for both predictors do not differ a lot. Even though differences can be observed in early iterations in particular, the step-lengths still have the same order of magnitude. This is not only the case for this example but overall in this simulation setup. Having this in mind, similar results for both approaches (FSL and ASL) are not very surprising anymore: there is hardly any difference in the approaches, since the updates do not need different step-lengths to be balanced. In the next subsection we will examine a case in which the data calls for different step-lengths, and see how both methods perform under those changed circumstances.

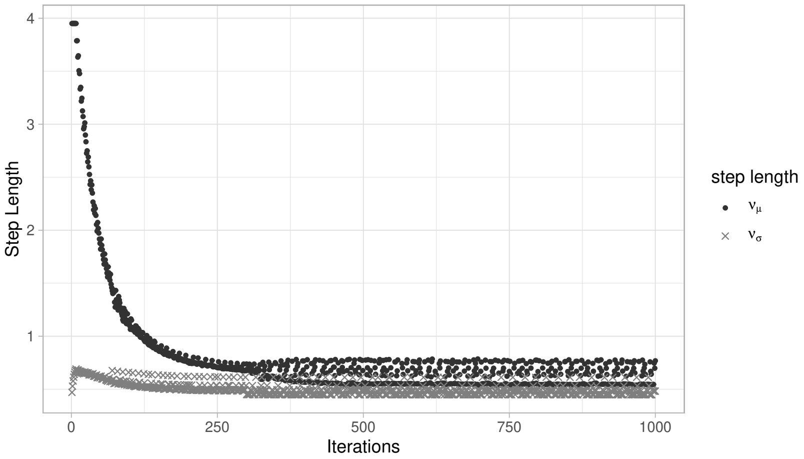


Figure 2.3 Comparison of the optimal step-lengths $\nu^{*[m]}_{j^*,\boldsymbol{\mu}}$ and $\nu^{*[m]}_{j^*,\boldsymbol{\sigma}}$ in SAASL from one of the 100 simulation runs. The step-lengths for $\boldsymbol{\mu}$ are in black dots, the step-lengths for $\boldsymbol{\sigma}$ in grey cross. Different horizontal layers of dots/crosses correspond to different covariates.

2.3.2 Large Variance with resulting Imbalance between Location and Scale

As discussed above, the Gaussian location and scale model in section 2.3.1 do not lead to a large difference between FSL and ASL, as the optimal step-lengths for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are roughly similar and the imbalance between the updates for the two predictors in FSL is thus not large. In this section, we investigate a setting with a large variance, which leads to a stronger imbalance between the two parts of the model.

In the following, we use SAASL as a representative of the adaptive approaches in our presentation, as it yields identical results to ASL, but avoids the numerical search for the optimal $\nu_{\boldsymbol{\mu}}$ by using the analytical result (2.5). Since estimated effects generally deviate more strongly from theoretical values than before due to the large variance (details will be discussed later), we additionally compare the results to those obtained using GAMLSS with penalized maximum likelihood estimation as implemented in the R-package `gamlss` (Rigby and Stasinopoulos, 2005).

Consider the data generating mechanism $y_i \sim \mathrm{N}(\mu_i, \sigma_i), i \in \{1, \cdots, 500\}$ with $B = 100$ simulation runs. The predictors are determined by

$$\begin{aligned} \eta_{\boldsymbol{\mu}}(\boldsymbol{x}_{i\cdot}) &= \mu_i = 1 + x_{i1} + 2x_{i2} - x_{i3} \\ \eta_{\boldsymbol{\sigma}}(\boldsymbol{x}_{i\cdot}) &= \log(\sigma_i) = 5 + 0.1x_{i1} - 0.2x_{i2} + 0.1x_{i3}, \end{aligned}$$

where $\boldsymbol{x}_{\cdot j} \sim \mathrm{U}(-1,1), j \in \{1,2,3,4,5\}$, $\boldsymbol{x}_{\cdot 4}$ and $\boldsymbol{x}_{\cdot 5}$ are noise variables. The choice of $\eta_{\boldsymbol{\sigma}}$ leads to an extremely large standard deviation in the order of 150 due to the large intercept 5. The stopping iteration is obtained by 10-fold CV, and the maximum number of iterations is 3000 and 2,000,000 for SAASL and FSL respectively. The main goal of this simulation setting is to highlight the imbalance problem of FSL when the scale parameter is large. As many noise variables will make it difficult to demonstrate the differences between FSL and adaptive approaches, we include only two noise variables in this example for illustration.

As can be seen in figure 2.4, both fixed and adaptive step-lengths yield reasonable estimates regarding $\eta_{\boldsymbol{\sigma}}$, but FSL results in many false negative estimates equal to zero for $\eta_{\boldsymbol{\mu}}$ in the majority of the simulation runs. This is of course connected to the relative importance of the variance component in this setting, which should in itself already lead to a preference for updating $\eta_{\boldsymbol{\sigma}}$ rather than $\eta_{\boldsymbol{\mu}}$ in early boosting iterations due to the fact that the negative gradient for $\boldsymbol{\mu}$ (i.e. $u_{\boldsymbol{\mu},i} = \sum_{i=1}^{n}(y_i - \mu_i)/\sigma_i^2$ with large σ_i) is actually scaled by the variance (recall the large intercept 5, log-link and the resulting exponential transformation) and hence very small. As a consequence, the impact on the global loss of base-learners fit to the gradient is also small compared to those suggested for updates regarding $\boldsymbol{\sigma}$ in step 11 of algorithm 1. Then, using the same step-length for both parameters makes it clearly harder to identify informative effects on $\boldsymbol{\mu}$ as they are trivialized in comparisons.

The adaptive step-lengths implemented in SAASL compensates for this disadvantage. Compared to the simulation result in the previous subsection, the estimates regarding $\eta_{\boldsymbol{\mu}}$ are less precise with large variability around true values. However, this is not a problem of SAASL but again the consequence of the large variance, obscuring the effects on the mean, and it is also encountered using the penalized maximum likelihood approach implemented in the `gamlss`-package (called GAMLSS in figure 2.4). The variability in the estimates is actually somewhat smaller than for GAMLSS due to the regularization inherent in the boosting approach. This is also illustrated in figure 2.5 in the pairwise comparison of the estimated coefficients for both methods, where SAASL leads to similar but slightly closer to zero estimates compared to the penalized maximum likelihood based method GAMLSS.

Interestingly, figure 2.4 also reveals that the inability to identify informative variables

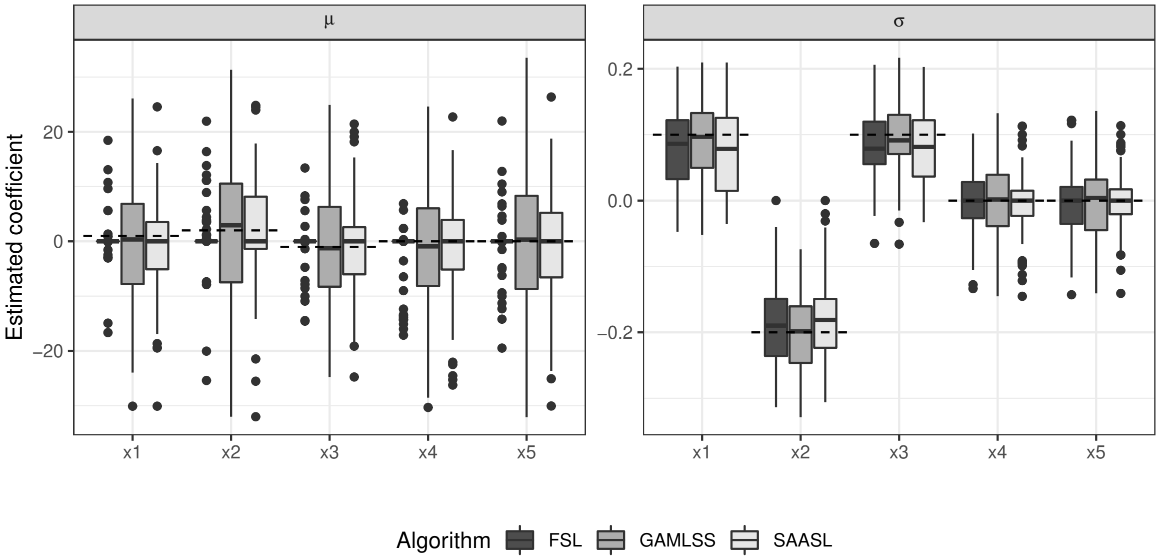


Figure 2.4 Distribution of coefficient estimates from $B = 100$ simulation runs. The true coefficients are marked by the dashed horizontal lines.

Table 2.1 Summary of the in-sample MSE for each estimation methods, i.e. $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
FSL	19848	21796	22547	22688	23579	27026
GAMLSS	19707	21687	22414	22586	23515	26883
SAASL	19679	21663	22372	22554	23443	26883

results in the lowest MSE for all three individual coefficients for $\boldsymbol{\mu}$ when using FSL (for more numerical details, see appendix A.3). As can be seen from table 2.1, however, the combined additive predictor performs worse in terms of overall MSE than both GAMLSS and SAASL, with the latter performing best.

To further highlight the differences in the selection behavior between FSL and SAASL, figure 2.6 illustrates the proportion of boosting iterations used to update $\boldsymbol{\mu}$ over the course of the model fits, i.e. $p_{m_{\boldsymbol{\mu}}} = m_{\boldsymbol{\mu}}/(m_{\boldsymbol{\mu}} + m_{\boldsymbol{\sigma}})$, where $m_{\boldsymbol{\mu}} + m_{\boldsymbol{\sigma}} = m_{\text{stop}}$. The bimodal distribution for FSL observed in the histogram in panel (a) demonstrates another problem of the fixed step-lengths in this setting. Considering many estimates equal or close to zero observed in figure 2.4, the mode close to $p_{m_{\boldsymbol{\mu}}} = 0$ is expected, as it describes the proportion of simulation runs where $\boldsymbol{\mu}$ has not been updated at all. However, as soon as at least one base-learner for $\boldsymbol{\mu}$ is recognized as an effective model parameter, the small step-length fixed at 0.1 requires a huge number of updates for the base-learner to actually make an impact on the global loss (hence the large number of

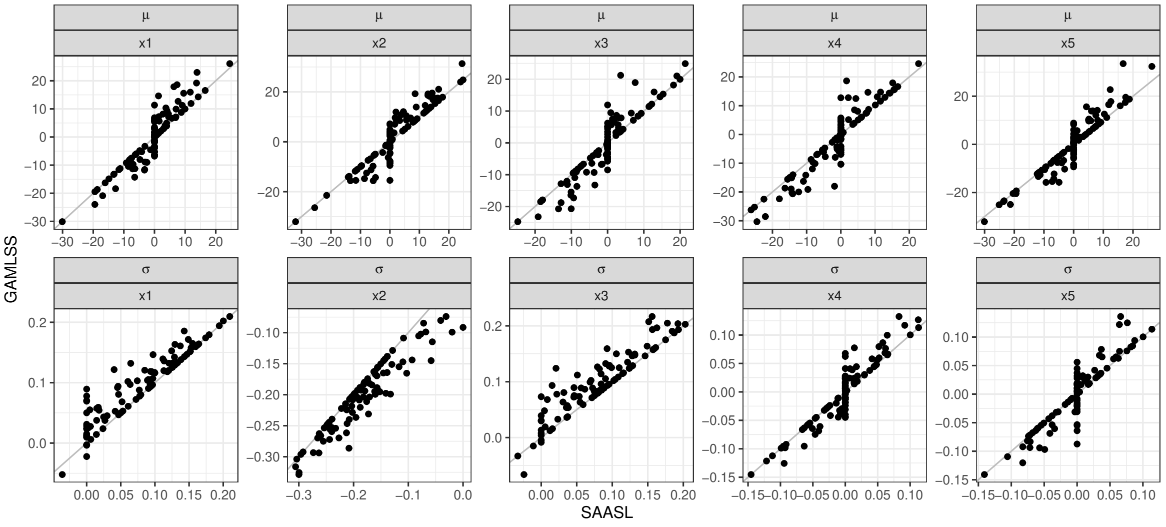


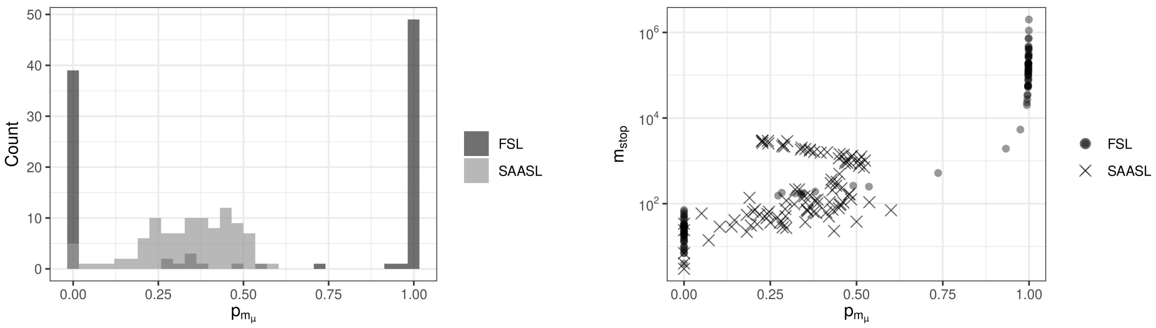
Figure 2.5 Pairwise comparison of the estimated coefficients between GAMLSS and SAASL for both model parameter $\boldsymbol{\mu}$ (top row) and $\boldsymbol{\sigma}$ (bottom row).

maximum iterations allowed for FSL). This results in the second mode around $p_{m_\mu} = 1$, as the algorithm is mainly occupied with μ in the corresponding runs.

This is also illustrated by the scatter plot in figure 2.6b, where p_{m_μ} is plotted against the stopping iteration m_{stop}. Note that the y-axis is displayed with a logarithmic scale and each tick on the y-axis represents a tenfold increase over the previous one. The few points (FSL), whose m_{stop} lie between 10^2 and 10^3, show a better balance between the updates of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ than other points, i.e. the middle region of p_{m_μ}. But we also observe a bimodal distribution for FSL, i.e. lots of points are equal or close to $p_{m_\mu} = 0$ and 1, with very low and extremely large values for m_{stop} resulting, respectively.

As for SAASL, we observe a unimodel distribution of p_{m_μ} in figure 2.6a. The mode smaller than 0.5 indicates SAASL updates $\boldsymbol{\sigma}$ a little more frequently than $\boldsymbol{\mu}$. Unlike the cyclical approach that enforces an equal number of updates for all distribution parameters, the balance formed by SAASL is more natural. This balance enables alternate updates between two predictors even though they lie on different scales. Therefore, the information in $\boldsymbol{\mu}$ can be fairly discovered in time and it reduces the risk of overlooking the informative base-learners with respect to $\boldsymbol{\mu}$. The number of simulations runs, in which $\boldsymbol{\mu}$ is not updated at all ($p_{m_\mu} = 0$), reduces from 39 in FSL to only 5 in SAASL. Moreover, none of the 100 simulations requires a substantial amount of updates for $\boldsymbol{\mu}$ to get well estimated coefficients (cf. also figure 2.4).

Table 2.2 displays the information about false positives and false negatives of the



(a) Histogram **(b)** Scatter plot

Figure 2.6 Distribution of $p_{m_{\boldsymbol{\mu}}}$ in $B = 100$ simulation runs. (a) Histogram of $p_{m_{\boldsymbol{\mu}}}$. The histogram of the two approaches are overlayed using transparency. (b) Scatter plot of m_{stop} against $p_{m_{\boldsymbol{\mu}}}$. Points and crosses are displayed with transparency. The y-axis is displayed on a logarithmic scale with base 10. Each tick represents a tenfold increase over the previous one.

Table 2.2 The number of simulations with false positives and false negatives for each variable under different modelling methods with respect to the two model parameters. The false negatives part shows the number of simulations in which the informative variables are excluded from the final model, and the false positives part shows how many simulations include the non-informative variables in their final model. Values are taken at the stopping iteration determined by 10-fold CV.

		False Negatives			False Positives	
		$\boldsymbol{x}_{\cdot 1}$	$\boldsymbol{x}_{\cdot 2}$	$\boldsymbol{x}_{\cdot 3}$	$\boldsymbol{x}_{\cdot 4}$	$\boldsymbol{x}_{\cdot 5}$
μ	FSL	83	77	81	21	20
	SAASL	28	24	28	72	73
σ	FSL	9	1	6	83	82
	SAASL	18	1	9	70	67

two approaches in all 100 simulations with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. For example, the second and fourth number 77 and 21 in the first line indicate that the informative variable $\boldsymbol{x}_{\cdot 2}$ is not included in the final model in 77 out of 100 simulation runs (i.e. false negative), while there are 21 simulations whose final model contains the non-informative variable $\boldsymbol{x}_{\cdot 4}$ (i.e. false positive). Similar as figure 2.2 in section 2.3.1, the conservative small step-length for $\boldsymbol{\mu}$ in FSL increases the number of boosting iterations, but reduces the risk of overfitting. Less simulations containing noise variables for $\boldsymbol{\mu}$ in FSL than in SAASL confirms this behavior. According to equation (2.11) the ASLs $\nu_{j^*,\boldsymbol{\sigma}}$ are a sequence of values around 0.05, and (except for the values at early boosting iterations) most of them smaller than 0.1. There are correspondingly slightly more simulations in FSL overfitting the $\boldsymbol{\sigma}$-submodel than in SAASL.

Although non-informative variables of $\boldsymbol{\mu}$ are excluded from the FSL model, the informative ones are excluded as well. Actually $\boldsymbol{\mu}$ is not updated in many simulations at all (cf. figure 2.6a). The false negatives part of table 2.2 for $\boldsymbol{\mu}$ confirms this. The informative variables $\boldsymbol{x}_{\cdot 1}$ to $\boldsymbol{x}_{\cdot 3}$ are excluded from the final model in the majority of simulations with FSL but not with SAASL. For $\boldsymbol{\sigma}$, the smaller step-length $\nu_{j^*,\boldsymbol{\sigma}}$ in SAASL selects variables more conservatively and as a consequence slightly more simulations underfit the $\boldsymbol{\sigma}$-submodel in SAASL than in FSL, but the difference is far less pronounced.

2.4 Applications

We apply the proposed algorithms to two datasets. The malnutrition dataset demonstrates the shortcomings of FSL and the pitfalls of using numerical determination of ASL with a fixed search interval, and with the riboflavin dataset we illustrate the variable selection properties of each algorithm.

2.4.1 Malnutrition of children in India

The first data called `india` from the R package `gamboostLSS` (Hofner et al., 2018; Fahrmeir and Kneib, 2011) are sampled from the Standard Demographic and Health Survey between 1998 and 1999 on malnutrition of children in India (Fahrmeir and Kneib, 2011). The data sample contains 4000 observations and four variables (BMI of the child (cBMI), age of the child in months (cAge), BMI of the mother (mBMI) and age of the mother in years (mAge)). The outcome of interest in this case is a numeric z-score for malnutrition ranging from -6 to 6, where the negative values represent malnourished children. To highlight the problem of using a fixed step-length, we work with the original variable stunting (corresponding to 100 * z-score). The identity and logarithm functions are used as link functions for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ respectively.

Because this is not a high-dimensional data example, we use the GAMLSS with penalized maximum-likelihood estimation as a gold standard to examine the effectiveness of adaptive approaches.

Table 2.3 lists the estimated coefficients of each variable on the predictors $\eta_{\boldsymbol{\mu}}$ and $\eta_{\boldsymbol{\sigma}}$ at the stopping iteration tuned by 10-folds CV, where the maximum number of iterations is set to 2000. The estimated intercept in $\eta_{\boldsymbol{\sigma}}$ indicates a large variance of the response, with the setting thus being similar to the second simulation above. It is therefore not surprising that FSL selects only one variable (cAge) for $\eta_{\boldsymbol{\mu}}$, i.e. a large number of updates for the base-learner are required but the given maximal boosting iteration is not large enough. In practice we can certainly increase the maximum number of iterations as well as enlarge the commonly applied step-length 0.1 in order to estimate the coefficients well. But their choices are very subjective and probably result in tedious manual fine-tuning based on trial and error.

The ASL method with the default predefined search interval $[0, 10]$ encounters a similar problem as FSL. Apart from the only selected and underfitted variable cAge for $\boldsymbol{\mu}$, the two variables (cBMI and cAge) for the $\boldsymbol{\sigma}$-submodel are also underfitted compared with the results from the gold standard GAMLSS. The reason for this phenomenon lies in the relationship between the variance and step-length discussed in equation (2.5). The log-link or exponential transformation for $\eta_{\boldsymbol{\sigma}}$ in this example data requires a sequence of huge step-lengths, but the default search interval does not fulfill this requirement.

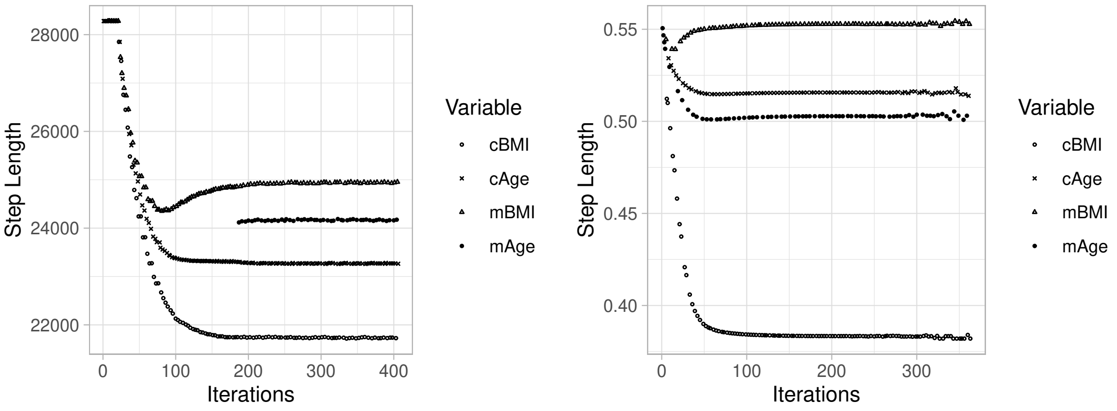
An estimation of ASL by increasing its search interval to $[0, 50000]$, denoted as ASL5 in table 2.3, results in coefficients comparable to those of GAMLSS. But choosing a suitable search interval becomes an unavoidable side task for ASL when analyzing this kind of dataset.

The results of the two semi-analytical approaches hardly differ from the maximum likelihood based GAMLSS. Unlike the numerical determination with a fixed search interval in ASL, the analytical approaches replace this procedure with a direct and precise solution that gets rid of the potential manual intervention (e.g. increasing the search interval). Contrary to the direct influence of the variance on $\nu^{*[m]}_{j^*,\boldsymbol{\mu}}$ in equation (2.5), the optimal step-length $\nu^{*[m]}_{j^*,\boldsymbol{\sigma}}$ is dominated by the chosen base-learner, but as the number of learning iterations increases, such effects gradually disappear, and $\nu^{*[m]}_{j^*,\boldsymbol{\sigma}}$ finally converges to 0.5. Thus, our default search interval $[0, 1]$ is sufficient for $\nu^{*[m]}_{j^*,\boldsymbol{\sigma}}$, and increasing the range of search interval as for $\nu^{*[m]}_{j^*,\boldsymbol{\mu}}$ in ASL is almost never necessary.

Theoretically, the ASL with a sufficiently large search interval (ASL5 in this example) and SAASL should result in the same values as discussed in the previous theoretical section. Due to the calculation accuracy of computers and the numerical optimization steps, their outputs are very similar but can differ slightly for the malnutrition data.

Table 2.3 Comparison of the estimated coefficients.

		FSL	ASL	ASL5	SAASL	SAASL05	GAMLSS
(Intercept)	η_{μ}	-174.772	-169.203	-91.160	-91.160	-91.160	-91.160
	η_{σ}	4.881	4.874	4.912	4.912	4.912	4.912
cBMI	η_{μ}	<0.001	<0.001	-13.925	-13.925	-13.925	-13.926
	η_{σ}	-0.003	-0.003	-0.015	-0.015	-0.015	-0.015
cAge	η_{μ}	-0.038	-0.371	-5.847	-5.847	-5.847	-5.847
	η_{σ}	-0.001	-0.001	0.003	0.003	0.003	0.003
mBMI	η_{μ}	<0.001	<0.001	11.708	11.708	11.708	11.708
	η_{σ}	0.009	0.009	0.009	0.009	0.009	0.009
mAge	η_{μ}	<0.001	<0.001	0.026	0.026	0.026	0.026
	η_{σ}	0.005	0.005	0.005	0.005	0.005	0.005



(a) $\nu^{*[m]}_{j^*,\boldsymbol{\mu}}$ **(b)** $\nu^{*[m]}_{j^*,\boldsymbol{\sigma}}$

Figure 2.7 The optimal step-length of each model parameters against the boosting iterations. Up to the stopping iterations specified by 10-folds CV (here $m_{\text{stop}} = 769$), 406 iterations are used to update $\boldsymbol{\mu}$ and 363 iterations are used to update $\boldsymbol{\sigma}$.

Figure 2.7 presents the optimal step-lengths $\nu^{*[m]}_{j^*,\boldsymbol{\mu}}$ and $\nu^{*[m]}_{j^*,\boldsymbol{\sigma}}$ using SAASL for each variable up to 769 boosting iterations specified by 10-fold CV for one simulation run. Apparently, the optimal step-lengths for $\boldsymbol{\mu}$ over the entire learning process are over 20000, which is far larger than the fixed step-length 0.1 and the upper boundary 10 of the predefined search interval in ASL. Without knowing this information, it is not trivial to determine the search interval for $\nu^{*[m]}_{j^*,\boldsymbol{\mu}}$. And we thus (after acquiring this graphic) re-estimated the example data with ASL5.

Additionally, figure 2.7b illustrates the optimal step-length for $\boldsymbol{\sigma}$. After several boosting iterations the optimal values of each covariate converge to their own stable regions (ranging from about 0.38 to 0.56). As discussed above, the optimal step-lengths for $\boldsymbol{\sigma}$ should be some values around 0.5, and this graphic confirms this statement.

As this example is not high-dimensional and does not necessarily require variable selection, we can use GAMLSS with penalized maximum likelihood estimation for comparison. The fact that its results are very similar to those of the semi-analytical approaches indicates that results from SAASL and SAASL05 are reliable. The only alternative to achieve balance between predictors would be using a cyclical algorithm (with the downsides discussed in the introduction). Rescaling the response variable or standardizing the negative partial derivatives could reduce the scaling problem to some extend, but would not eliminate the need to increase the step-length or reduce the imbalance between predictors.

2.4.2 Riboflavin dataset

This data set describes the riboflavin (also known as vitamin B_2) production by Bacillus subtilis, containing 71 observations and 4088 predictors (gene expressions) (Bühlmann et al., 2014; Dezeure et al., 2015). The log-transformed riboflavin production rate, which is close to a Gaussian distribution, is regarded as the response. This data set is chosen to demonstrate the capability of the boosting algorithm to deal with situations in which the number of covariates exceeds the number of observations. Please note that a comparison to the original GAMLSS algorithm is not possible in this case, since the algorithm is not able to deal with more model parameters than available observations. In order to compare the out-of-sample MSE of each algorithm, we select 10 observations randomly as the validation set.

Table 2.4 summarize the selected informative variables for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ separately at the stopping iteration tuned by 5-fold CV, the corresponding coefficients are listed in appendix A.4. The results in both tables demonstrate the intersection of the selected variables, for example FSL selects 13 informative variables in total, and 9 of them are also chosen by ASL and SAASL, and there are 11 variables common with SAASL05. In general, for both $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, more variables are included in the adaptive approaches and the difference in the selected variables mainly lies between the adaptive and fixed approach. Because the optimal step-length $\nu_{j^*,\boldsymbol{\mu}}^{*[m]}$ lies in the predefined search interval $[0, 10]$ (and is actually smaller than 1, i.e. the adaptive step-length $\nu_{j^*,\boldsymbol{\mu}}^{[m]} < 0.1$), and $\nu_{j^*,\boldsymbol{\sigma}}^{*[m]}$ lies also in a narrower predefined search interval $[0, 1]$, ASL and SAASL have the same results. Moreover, as the adaptive step-length is smaller than the fixed step-length 0.1,

Table 2.4 Number of chosen variables for η_{μ} and η_{σ}. The diagonal depicts the number per method, the off-diagonal elements overlapping variables.

	η_{μ}				η_{σ}			
	FSL	ASL	SAASL	SAASL05	FSL	ASL	SAASL	SAASL05
FSL	13	9	9	11	16	9	9	12
ASL	9	20	20	18	9	17	17	15
SAASL	9	20	20	18	9	17	17	15
SAASL05	11	18	18	24	12	15	15	24

Table 2.5 Comparison of the out-of-sample MSE.

	FSL	ASL	SAASL	SAASL05	`glmnet`
MSE	2.611	1.111	1.111	1.193	0.946

the adaptive approaches make conservative (small) updates, leading to more boosting iterations. Several of gene expressions for $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are selected by all algorithms and are thus consistently included in the set of informative covariates. Actually almost all gene expressions chosen by FSL are also recognized as informative variables by all other methods.

To compare the performance of each algorithm, table 2.5 lists the out-of-sample MSE. In contrast to the fixed approach, the three adaptive approaches perform in general well, where the performance of SAASL05 is slightly worse than the other two. In addition, table 2.5 demonstrates also the result of Lasso estimator from the R package `glmnet` (Friedman et al., 2010) suggested by Bühlmann et al. (2014). The mean squared prediction error of `glmnet` is the smallest among the five approaches, but the difference with the adaptive approaches is relatively small.

As `glmnet` cannot model the scale parameter $\boldsymbol{\sigma}$, only the estimated coefficients of the $\boldsymbol{\mu}$-submodel are provided in appendix A.4. Out of the 21 genes selected by `glmnet`, 7 and 9 of them are common with the ASL/SAASL and SAASL05, respectively. The signs (positive/negative) of the estimated coefficients of these common covariates from `glmnet` match the adaptive approaches. This comparison indicates that the boosted GAMLSS with adaptive step-length is an applicable and competitive approach for high-dimensional data analysis.

2.5 Conclusions and Outlook

The step-length is often not treated as an important tuning parameter in many boosting algorithms, as long as it is set to a small value. However, if complex models like GAMLSS with several predictors for the different distribution parameters are estimated, different scales of distribution parameters can lead to imbalanced updates and resulting bad performances if one common small fixed step-length is used, as we show in this chapter.

The main contribution of this chapter is the proposal to use separate adaptive step-lengths for each distribution parameter in a non-cyclical boosting algorithm for GAMLSS. In addition to the resulting balance in updates between different distribution parameters, a balance between over- and underfitting is obtained by taking only a proportion (shrinkage parameter) such as 10% of the determined optimal step-length as the adaptive step-length. The optimal step-length can be found by optimization procedures such as a line search. We illustrated with an example the importance of updating the search interval for the search if necessary to find the optimal solution.

For Gaussian location and scale models, we derived an analytical solution for the adaptive step-length for the mean parameter $\boldsymbol{\mu}$, which avoids numerical optimization and specification of a search interval. For the scale parameter $\boldsymbol{\sigma}$, we obtained an approximate solution of 0.5 (or 0.05 with 10% proportion), which gives a better motivated default value than 0.1 relative to the step-length for $\boldsymbol{\mu}$, and discussed a combination with a one-dimensional line search in the semi-analytical approach.

In simulations and empirical applications, we showed favorable behavior compared to use a fixed step-length FSL. We showed highly competitive results of our adaptive approaches compared to a standard GAMLSS with respect to estimation accuracy for the low-dimensional case, while the adaptive boosting approach has the advantages of shrinkage and variable selection, which make it also applicable to the high-dimensional case of more covariates than observations. Overall, the semi-analytical method for adaptive step-length selection performs best among the considered methods.

In this chapter we focus on the Gaussian location and scale models to derive analytical or semi-analytical solutions for the optimal step-length, but in most cases, a line search has to be conducted for all distribution parameters. In the future, if possible it is worth investigating analytical adaptive step-lengths for other distributions, because analytical or approximate adaptive step-lengths increase the numerical efficiency and also reveal the relationships between the optimal step-lengths for different parameters and model parameters (as well as properties of commonly used but probably less than ideal step-length settings).

We are confident that the adaptive step-length concept is relevant way beyond the Gaussian specification, so further work should contain the study on the stability and effectiveness of the implementation of adaptive step-length to other common distributions or zeor-inflated count models. Further work should also include the implementation of further (e.g. non-linear, spatial etc.) effects (Hothorn et al., 2011) into the model, and test the influence of the adaptive step-length on such effects. Moreover, we discovered correlations between the optimal step-length $\nu_{j^*,\boldsymbol{\mu}}^{*[m]}$ of a variable and the coefficient of this variable in the $\boldsymbol{\sigma}$-submodel through our application of the algorithm. Future work should also investigate the relationship among the optimal step-lengths of different parameters and the relationship of these step-lengths to model coefficients.

Chapter 3

Bayesian Learners in Gradient Boosting for Linear Mixed Models

Linear mixed models (LMM) (Laird and Ware, 1982) are widely used in longitudinal data analysis as they incorporate random effects to deal with group-specific heterogeneity. Data involving repeated observations of the same variables are common in epidemiology, medical statistics and many other fields.

Likelihood-based methods are often used to make inference for (generalized) linear mixed models (Bates et al., 2000; Gumedze and Dunne, 2011). These kinds of estimations based on the maximum likelihood theory depend on the correct specification of the distribution of estimators, since for misspecified distributions (of the random effects), the estimators will be inconsistent and biased (Heagerty and Kurland, 2001; Litière et al., 2008). To address the bias problems, especially with binary data or correlated random effects (Breslow and Clayton, 1993; Breslow and Lin, 1995; Lin and Zhang, 1999), a fully Bayesian inference via Markov Chain Monte Carlo (MCMC) simulation is proposed by Fahrmeir and Lang (2001) for generalized additive and semiparametric mixed models. One advantage of a Bayesian approach is that it is easier to take uncertainty in variance components into account (Zhao et al., 2006). The Bayesian inference is also suggested by Fong et al. (2010) due to the unreliability of the likelihood-based inference with variance components being difficult to estimate, especially for small sample sizes, but they replaced the compute-intensive MCMC simulation with the much faster integrated nested Laplace approximation (INLA).

Regarding regularization and variable selection, the $L1$-penalized estimation (Lasso) is integrated into the likelihood-based inference (Schelldorfer et al., 2011; Groll and Tutz, 2014), which enables the method to deal with high-dimensional data. A likelihood-based boosting approach for fitting generalized linear mixed models (GLMMs) is presented in Tutz and Groll (2010) for the first time, that integrates the boosting technique into the mixed model estimation. Especially with focus on penalized likelihood inference, likelihood-based boosting (Tutz and Binder, 2006) represents an alternative to the well known gradient boosting technique (Friedman, 2001) and is due to its componentwise maximization routine (Bühlmann and Yu, 2003; Hothorn et al., 2010) suitable for variable selection and high-dimensional data structures. The R package `gamboostLSS` (Hofner et al., 2016) shows the latest progress of applying boosting framework to additive models, but it cannot solve the problem of estimation bias and imbalanced choice between fixed and random effects in LMMs. The potential bias of the likelihood-based boosting estimators occurring in the presence of cluster-constant covariates like gender or treatment group in longitudinal studies is addressed in Griesbach et al. (2021a) and they proposed an improved algorithm where the random effects show no correlation with any observed variables. Moreover, this bias correction was successfully adapted to gradient boosting estimating technique for linear mixed models Griesbach et al. (2021b). However, variable selection regarding the random structure in their approach is not allowed, as the random effects have to be specified in advance.

While boosting provides a very flexible model-based inference, there is no straightforward way to conduct standard parametric hypothesis tests. The biased estimate induced by shrinkage affects also other viable alternatives such as bootstrap confidence intervals (Hepp et al., 2019; Mayr et al., 2017b). However, with Bayesian sampling, the variability of the coefficients (e.g. credible intervals) can be fused as part of the estimation procedure. Therefore, in this chapter, we introduce a novel inference method that integrates a Bayesian learner into gradient boosting in linear mixed models, denoted as `BayesBoost`, which incorporates the two concepts, Bayesian statistics and boosting, for the first time and benefits from the shrinkage and variable selection properties of boosting and from the uncertainty estimates of Bayesian inference.

The `BayesBoost` method divides the estimation procedure into two parts, the componentwise gradient boosting estimation for the fixed effects and the merged estimation (boosting with a Bayesian learner) for the random effects. The automatic selection of the random effects is based on their contribution to the model measured e.g. by the in-sample mean squared error (MSE), and it assumes that only informative fixed effects have random effects, in other words, only when a covariate is selected as a fixed effect, it can be considered for the choice of random effects. Thus, the selection of random effects is involved inside the selection of fixed effects conducted by the boosting technique, and is performed with the estimation simultaneously. We also provide the interface for a fixed user-defined random effects structure or a flexible user-defined random effect candidates set to make the automatic selection without any assumptions.

However, the usage of Bayesian inference makes the model improvement being affected by the stochasticity of MCMC samples. For a common boosting algorithm like gradient boosting, model evaluation (such as MSE) decreases monotonously as the boosting iteration increases, but the implementation of a Bayesian learner or a MCMC procedure in `BayesBoost` cannot guarantee this descending order between adjacent iterations. This makes the widely used Akaike information criterion (AIC) (Akaike, 1973) or the more suitable conditional AIC (cAIC) (Vaida and Blanchard, 2005; Liang et al., 2008; Greven and Kneib, 2010) for mixed models difficult to serve as the stopping criterion due to the stochasticity of the global minimum. We therefore suggest to use probing Thomas et al. (2017) to prevent overfitting, which is a variable selection method based on the addition of randomly permuted variables.

The chapter is structured as follows: Section 3.1 specifies the linear mixed models and introduces how to make an inference with the `BayesBoost` algorithm, and discussions on random effects selection and model choice are also covered. Section 3.2 evaluates the performance of the `BayesBoost` algorithm in three different simulation scenarios and highlights its new features. Section 3.3 presents the application of the `BayesBoost` algorithm to the riboflavin data set, which deals with the relationship between gene expression and the riboflavin production by Bacillus subtilis. At last, Section 3.4 concludes with a discussion and outlook.

3.1 Methods

This section starts with the specification of linear mixed models for longitudinal and clustered data. Then we propose the `BayesBoost` algorithm with a detailed explanation of the parameter estimation. The model selection criterion based on probing is also briefly introduced in the end of this section.

3.1.1 Model specification

For clusters or individuals $i = 1, \dots, m$ with $n = \sum_{i=1}^{m} n_i$, where n_i denotes the replicates of the i-th individual, consider the linear mixed model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\gamma} + \boldsymbol{\varepsilon} \tag{3.1}$$

with

$$\begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\varepsilon} \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \boldsymbol{G} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R} \end{pmatrix} \right), \tag{3.2}$$

where $\boldsymbol{y} = (\boldsymbol{y}_1, \dots, \boldsymbol{y}_m)^T$, $\boldsymbol{\varepsilon} = (\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_m)^T$ and $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_m)^T$, as well as the design matrices $\boldsymbol{X} = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_m)^T$ and $\boldsymbol{Z} = \text{blockdiag}(\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_m)$. More specifically, $\boldsymbol{y}_i$ is the n_i-dimensional vector of responses for individual i, $\boldsymbol{X}_i$ and $\boldsymbol{Z}_i$ are $n_i \times (p+1)$- and $n_i \times (q+1)$-dimensional design matrices constructed from known covariates, $\boldsymbol{\beta}$ is a $(p+1)$-dimensional vector of fixed effects with intercept, $\boldsymbol{\gamma}_i$ is a $(q+1)$-dimensional vector of cluster-specific random effects with random intercept, and $\boldsymbol{\varepsilon}_i$ is a n_i-dimensional vector of errors.

We assume independency of $\boldsymbol{\gamma}$ and $\boldsymbol{\varepsilon}$ with positive definite covariance matrices. The covariance matrices $\boldsymbol{G}$ and $\boldsymbol{R}$ are block-diagonal with

$$\begin{aligned} \boldsymbol{R} &= \text{blockdiag}(\sigma^2 \boldsymbol{\Sigma}_{n_1}, \cdots, \sigma^2 \boldsymbol{\Sigma}_{n_m}), \\ \boldsymbol{G} &= \text{blockdiag}(\boldsymbol{Q}, \cdots, \boldsymbol{Q}), \end{aligned}$$

where $\boldsymbol{\gamma}_i \sim \text{N}(\mathbf{0}, \boldsymbol{Q})$ with $(1+q) \times (1+q)$-covariance matrix $\boldsymbol{Q}$. For i.i.d. errors, which is also the case in this chapter, $\boldsymbol{R}$ simplifies to $\boldsymbol{R} = \sigma^2 \boldsymbol{I}$.

The predictor $\boldsymbol{\eta}$ for the response displays as

$$\boldsymbol{y} = \boldsymbol{\eta} = \sum_{k=1}^{p} \boldsymbol{\eta}_k$$

with

$$\boldsymbol{\eta}_k = \boldsymbol{X}_k\boldsymbol{\beta}_k + \boldsymbol{Z}_k\boldsymbol{\gamma}_k, \tag{3.3}$$

where $\boldsymbol{X}_k = (1, X_k)$ is a design matrix that accounts for the intercept and the corresponding parameter vector is $\boldsymbol{\beta}_k = (\beta_0, \beta_k)$. $\boldsymbol{Z}_k$ and $\boldsymbol{\gamma}_k$ represent the design matrix constructed by the covariate X_k and its random effect respectively.

3.1.2 Bayesian boosting inference method

This chapter proposes a novel estimation and variable selection method for linear mixed effects models by integrating Bayesian inference, which constructs a Bayesian learner for the precision of parameters, into the boosting framework, which is famous for its variable selection and shrinkage features. In this chapter we refer to this method as `BayesBoost`.

The additive representation (3.3) makes our approach possible as it naturally divides the predictor into fixed and random parts and each part can be estimated separately by treating the others as an offset. Specifically, the `BayesBoost` estimation procedure contains two steps: the first step is to estimate fixed effects through gradient boosting as usual, and the second step is to make Bayesian inference by employing a Bayesian learner to the random effects while treating the estimated fixed effects as offsets.

Another emphasis is the selection of random effects. `BayesBoost` assumes that only variables with an already selected fixed effect should be given the opportunity to obtain an additional random effect. This means the random effect is considered informative when its associated variable is already selected to have a fixed effect and the model improvement for its random effect is greater than its fixed effect.

We first provide a full description of `BayesBoost` (algorithm 4). Note that the presented description does not contain the stopping mechanism, which, however, is easy to be implemented. Details of stopping method as well as other important steps will be discussed in more detail in the following subsections.

Algorithm 4 `BayesBoost` for LMM

```
Initialize $\hat{\boldsymbol{y}}, i \in \{1, \dots, m\}, \hat{\beta}_0^{[0]}, \boldsymbol{R}^{(0)}, \boldsymbol{Q}^{[0]}, \Lambda_0^{[0]}, a, b, v_0$ and random set $E = \{\text{ranInt}\}$ containing only the random intercept.
Construct correction matrix $\boldsymbol{Z}$.
for $s = 1, \dots, s_{\text{stop}}$ do
    Compute the negative gradient vector
        $\boldsymbol{u}^{[s]} = -\frac{\partial}{\partial \boldsymbol{\eta}} \rho(\boldsymbol{y}, \boldsymbol{\eta}) \Big|_{\boldsymbol{\eta} = \hat{\boldsymbol{\eta}}^{[s-1]}}$
    Fit every covariate $X_k, k \in \{1, \dots, p\}$, separately to the gradient $\boldsymbol{u}^{[s]}$ with a regression model to get the coefficients $\hat{\boldsymbol{\beta}}_k$.
    Select the best-fitting base-learner that results in the least residual sum of squares
        $\text{MSE}_{k,\text{fixed}} = \frac{1}{n} \sum_{i=1}^{n} \left( u_i - \nu \boldsymbol{X}_{ik} \hat{\boldsymbol{\beta}}_k \right)^2, \quad \text{for } k^* = \underset{k \in \{1, \dots, p\}}{\arg\min} \text{MSE}_k.$
    if $X_{k^*}$ is not in the random effects set $E$ then
        Construct a potential design matrix $\boldsymbol{Z}_{\text{pot}}^{[s]}$, update $\boldsymbol{Q}_{\text{pot}}^{[s]}$, $\boldsymbol{G}^{(0)}$ and $\boldsymbol{\Lambda}_{0,\text{pot}}^{[s]}$.
    end if
    Draw MCMC samples for the random effect $\gamma^{(t)}$, variance $\sigma^{2(t)}$ and covariance $\boldsymbol{Q}^{(t)}$.
    Compute the posterior modes $\hat{\gamma}_{\text{mode, pot}}^{[s]}$, $\hat{\sigma}_{\text{mode}}^{2[s]}$ and $\hat{\boldsymbol{Q}}_{\text{mode, pot}}^{[s]}$.
    Calculate the model improvement regarding the random effect of $X_{k^*}$,
        $\text{MSE}_{k^*,\text{random}} = \frac{1}{n} \left( u_i - \boldsymbol{Z}_{ik^*,\text{pot}}^{[s]} \hat{\gamma}_{k^*,\text{mode, pot}}^{[s]} \right)^2$
    if $\text{MSE}_{k^*,\text{fixed}} < \text{MSE}_{k^*,\text{random}}$ then
        Reject the potential structure and reset them to the previous state.
    else
        Accept the potential structure and update random effects
    end if
end for
```

Fixed effects estimation

From a `BayesBoost` perspective, only random effects are considered as random variables, while fixed effects are assumed to be constant. So unlike in conventional full Bayesian inference, where the fixed effects vector $\boldsymbol{\beta}$ is obtained by MCMC simulation, it is updated by a boosting step in each iteration of the `BayesBoost` approach. In componentwise gradient boosting, the negative gradient vector

$$\boldsymbol{u}^{[s]} = \frac{\partial \rho(\boldsymbol{y}, \hat{\boldsymbol{\eta}}^{[s-1]})}{\partial \boldsymbol{\eta}} = \boldsymbol{y} - \hat{\boldsymbol{\eta}}^{[s-1]}, \quad s = 1, \ldots, m_{\text{stop}},$$

with the $L2$-loss $\rho(\cdot)$, i.e. $\rho(a, b) = \frac{1}{2}\sum(a-b)^2$, in boosting iteration s is fitted by each base-learner, yielding

$$\boldsymbol{u}^{[s]} \overset{\text{base-learner}}{\longrightarrow} \hat{h}(\boldsymbol{X}_k) \quad \text{for} \quad k = 1, \ldots, p.$$

For a linear base-learner, its correspondence to the model term shall be $\hat{h}(\boldsymbol{X}_k) = \boldsymbol{X}_k\hat{\boldsymbol{\beta}}_k$. Note that $\boldsymbol{X}_k = (1, X_k)$ and $\boldsymbol{\beta}_k = (\beta_0, \beta_k)$. The best-fitting base-learner is then selected based on the residual sum of squares with respect to $\boldsymbol{u}^{[s]}$

$$k^* = \underset{k \in \{1,\ldots,p\}}{\arg\min} \sum_{i=1}^{n} (u_i^{[s]} - \hat{h}(X_{ik}))^2,$$

where X_{ik} denotes the i-th observation of X_k.

The updated fixed effects with respect to the best-fitting covariate are

$$\hat{\boldsymbol{\beta}}^{[s]} = \hat{\boldsymbol{\beta}}^{[s-1]} + \nu \hat{\boldsymbol{\beta}}_{k^*}^{[s]}, \tag{3.4}$$

where ν denotes a step-length or learning rate. Since $\hat{\boldsymbol{\beta}}^{[s-1]}$ is a vector of length $(p+1)$ and $\hat{\boldsymbol{\beta}}_{k^*}^{[s]}$ is of length 2, updates of $\hat{\boldsymbol{\beta}}^{[s]}$ in equation (3.4) happen only in the intercept term and the corresponding k^*-th covariate, while all the other covariates remain unchanged.

Model improvement benefiting from the fixed effect X_{k^*} is measured with the MSE

$$\text{MSE}_{k^*,\text{fixed}} = \frac{1}{n}\sum_{i=1}^{n} \left(u_i - \nu \boldsymbol{X}_{ik^*}\hat{\boldsymbol{\beta}}_{k^*}^{[s]}\right)^2. \tag{3.5}$$

This serves later for deciding whether X_{k^*} should also contribute to the model from a random effect perspective, i.e. the selection of random effects.

Random effects estimation

After obtaining the estimated fixed effects, the full Bayesian inference for the parameters of interest is based on the posterior distribution

$$p(\boldsymbol{\gamma}, \boldsymbol{G}, \boldsymbol{R}|\tilde{\boldsymbol{y}}) \propto p(\tilde{\boldsymbol{y}}|\gamma, \boldsymbol{G}, \boldsymbol{R})p(\gamma|\boldsymbol{G})p(\boldsymbol{G})p(\boldsymbol{R}), \tag{3.6}$$

with $\tilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{[s]}$ treating fixed effects as an offset term and $\boldsymbol{\gamma}|\boldsymbol{G}$, $\boldsymbol{R}$ and $\boldsymbol{G}$ are assumed to be independent. In general, the posterior (3.6) cannot be displayed in a closed form, such that the full Bayesian inference is usually conducted through MCMC simulation, or more precisely through Gibbs sampler in this chapter.

The random effects distribution $\boldsymbol{\gamma} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{G})$ can be seen as a prior for random effects. Then, the full conditional distribution of $\boldsymbol{\gamma}$ is

$$\begin{aligned} p(\gamma|\tilde{\boldsymbol{y}}, \boldsymbol{G}, \boldsymbol{R}) \propto & p(\tilde{\boldsymbol{y}}|\gamma, \boldsymbol{G}, \boldsymbol{R})p(\gamma|\boldsymbol{G}) \\ & \propto \exp\left(-\frac{1}{2}\left(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\gamma\right)^T \boldsymbol{R}^{-1}\left(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\gamma\right)\right) \exp\left(-\frac{1}{2}\gamma^T \boldsymbol{G}^{-1}\gamma\right). \end{aligned} \tag{3.7}$$

The inner term of the exponential function in equation (3.7) is a sum of two squared forms, so the conditional distribution is Gaussian with parameters

$$\begin{aligned} \boldsymbol{\Sigma}_\gamma &= \left(\boldsymbol{Z}^T\boldsymbol{R}^{-1}\boldsymbol{Z} + \boldsymbol{G}^{-1}\right)^{-1}, \\ \boldsymbol{\mu}_\gamma &= \boldsymbol{\Sigma}_\gamma\left(\boldsymbol{Z}\boldsymbol{R}^{-1}\tilde{\boldsymbol{y}}\right). \end{aligned}$$

According to the model specification, the covariance matrix for i.i.d. errors $\boldsymbol{R} = \sigma^2\boldsymbol{I}$ is dominated by the hyperparameter σ^2. Moreover, a weakly informative inverse gamma prior $\sigma^2 \sim \mathrm{IG}(a, b)$ with small a and b is commonly proposed (Fahrmeir et al., 2021), because for $a = b$ and both values approaching zero, the distribution of $\log\sigma^2$ tends to be a uniform distribution. Thus, small values for a and b are identified with a weakly informative or noninformative prior. The full conditional density of σ^2 turns out to be

$$\begin{aligned} p(\sigma^2|\tilde{\boldsymbol{y}}, \gamma) &\propto p(\tilde{\boldsymbol{y}}|\boldsymbol{\gamma}, \sigma^2)p(\sigma^2) \\ &\propto (\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\boldsymbol{\gamma})^T(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\boldsymbol{\gamma})\right) \cdot (\sigma^2)^{-a-1} \exp\left(-\frac{1}{\sigma^2}b\right) \\ &= (\sigma^2)^{-\frac{n}{2}-a-1} \exp\left(-\frac{1}{\sigma^2}\left(b + \frac{1}{2}(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\boldsymbol{\gamma})^T(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\boldsymbol{\gamma})\right)\right), \end{aligned}$$

which is again an inverse gamma distribution $\mathrm{IG}(\tilde{a}, \tilde{b})$ with

$$\begin{aligned}\tilde{a} &= a + \frac{n}{2},\\ \tilde{b} &= b + \frac{1}{2}\left(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\boldsymbol{\gamma}\right)^T \left(\tilde{\boldsymbol{y}} - \boldsymbol{Z}\boldsymbol{\gamma}\right).\end{aligned}$$

The last parameter whose prior needs to be specified is the covariance matrix $\boldsymbol{G}$ of random effects. Analogously, the block-diagonal matrix $\boldsymbol{G} = \mathrm{blockdiag}(\boldsymbol{Q}, \ldots, \boldsymbol{Q})$ is dominated by the covariance matrix of single individual $\boldsymbol{Q}$. Usually, we assume an inverse Wishart prior for the covariance matrix, i.e. $\boldsymbol{Q} \sim \mathrm{IW}(v_0, \boldsymbol{\Lambda}_0)$, which can be understood as the multivariate case of an inverse gamma distribution. Recalling that $\boldsymbol{Q}$ is a $(1+q) \times (1+q)$-dimensional matrix and there exists totally m clusters, we have

$$\begin{aligned} p(\boldsymbol{Q}|\tilde{\boldsymbol{y}}, \boldsymbol{\gamma}, \boldsymbol{R}) &\propto p(\tilde{\boldsymbol{y}}|\boldsymbol{\gamma}, \boldsymbol{Q}, \boldsymbol{R}) p(\boldsymbol{\gamma}|\boldsymbol{Q}) p(\boldsymbol{Q}) \propto p(\boldsymbol{\gamma}|\boldsymbol{Q}) p(\boldsymbol{Q})\\ &\propto |\boldsymbol{Q}|^{-\frac{m}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{m} \boldsymbol{\gamma}_i^T \boldsymbol{Q}^{-1} \boldsymbol{\gamma}_i\right) \cdot |\boldsymbol{Q}|^{-\frac{v_0+(1+q)+1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{Q}^{-1}\boldsymbol{\Lambda}_0\right)\right)\\ &= |\boldsymbol{Q}|^{-\frac{m+v_0+(1+q)+1}{2}} \exp\left(-\frac{1}{2}\mathrm{tr}\left(\boldsymbol{Q}^{-1}\left(\boldsymbol{\Lambda}_0 + \boldsymbol{\gamma}^T\boldsymbol{\gamma}\right)\right)\right), \end{aligned}$$

where $|\cdot|$ denotes the determinant of a matrix. Therefore, the full conditional for $\boldsymbol{Q}$ is an inverse Wishart distribution with

$$\begin{aligned} v &= v_0 + m,\\ \boldsymbol{\Lambda} &= \boldsymbol{\Lambda}_0 + \boldsymbol{\gamma}^T\boldsymbol{\gamma}. \end{aligned}$$

Bayesian inference for these unknown parameters is then made according to the corresponding Gibbs samples drawn from the full conditional distributions. In conventional Bayesian inference for linear mixed models, the MCMC simulation procedure is performed only once. However, the combination of Bayesian inference and boosting makes it necessary that the procedure is performed the same number of times as the total of boosting iterations since the estimated fixed effects are treated as an offset term and each update yields a different $\tilde{\boldsymbol{y}}$ requiring the sampling process to be repeated. To minimize the invalid samples (burn-in) to the most extent, we can use the posterior modes from the last iteration instead of repeatedly using the initialized values as starting values of the MCMC process for the current iteration. The preference to the mode instead of mean is due to the asymmetric prior distribution of σ^2 and $\boldsymbol{Q}$, and the sample outliers affect the mean but have little effect on the mode. We omit burn-in of the individual MCMC chains for two reasons: the first one is that, due to the gradual convergence to the correct values the starting values in the later boosting iterations are already in the correct area, since they are based on the results of the previous boosting iteration. Hence, we could consider the whole chain generated by all boosting iterations except the last as pseudo burn-in. The second reason is the stability we get from the parameters generated solely by the boosting mechanism: the full conditional distributions are derived on constant fixed effects $\hat{\boldsymbol{\beta}}^{[s]}$, which also leads to less instability than seen in full MCMC approaches. There won't be large changes in the burn-in period between the samples as seen in conventional Bayesian inference, where $\boldsymbol{\beta}$ is regarded a random variable and must be drawn as well, because the process of approaching the stationary region is omitted.

Another point that deserves mentioning here is the nearest positive definite matrix. Even though model selection discussed in this chapter is based on probing (explanation see below), it is sometimes useful to analyze the AIC or cAIC, which requires a Cholesky decomposition of the covariance matrix $\boldsymbol{G}$ or just $\boldsymbol{Q}$ to avoid calculating the high-dimensional inverse matrix (Säfken et al., 2021). The covariance matrix $\boldsymbol{Q}$ constructed from the elementwise posterior mode of Gibbs samples, however, occasionally does not guarantee to be a positive definite matrix, in other words, the condition of Cholesky decomposition is not fulfilled. In practice, for the case of non-positive definite $\boldsymbol{Q}$, we suggest to transform it to its nearest positive definite matrix. The transforming algorithm is beyond the scope of this chapter, for more details please refer to Higham (2002).

Bias correction

A common problem of the likelihood-based boosting approach is the correlation between random intercepts and cluster-constant covariates (e.g. gender or age-group), and the problem can be addressed by introducing an additional correction step for the random effects within a usual boosting framework Griesbach et al. (2021a). But unlike the "piecewise" updates of random effects in usual boosting algorithms, `BayesBoost` extracts the whole information of random effects from residuals (i.e. $\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{[s]}$) at once in every boosting iteration through MCMC simulation, and this mechanism often induces ineffective updates of fixed effects. Due to stepwise build up of the boosting approach fixed effects explain only a little variance of the response in the beginning iterations, such that residuals contain lots of information, which ought to belong to the fixed effects part, but are explained by the random effects altogether. Consequently, the information extracted by fixed effects in the next boosting iteration from the remaining residuals $\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{[s-1]} - \boldsymbol{Z}\hat{\boldsymbol{\gamma}}^{[s-1]}$ becomes ineffective since lots of information have already been accounted by the random part.

Therefore, weakened and disentangled updates are meaningful not only for the cluster-constant covariates as emphasized by Griesbach et al. (2021a), but also for the cluster-varying variables in the `BayesBoost` framework. To prevent such correlation between fixed and random effects, we replace the original design-matrix of the random effects $\tilde{\boldsymbol{Z}}$ with

$$\boldsymbol{Z} = (\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T)\tilde{\boldsymbol{Z}}, \tag{3.8}$$

where $\boldsymbol{I} - \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ is known as the residual maker matrix, which is interpreted as a matrix that produces least squares residuals in the regression of $\tilde{\boldsymbol{Z}}$ on $\boldsymbol{X}$ when it multiplies any $\tilde{\boldsymbol{Z}}$. Note that the design matrix $\boldsymbol{Z}$ used in the entire algorithm including what we have discussed above is actually the corrected version (3.8). Based on the linear regression theory it can be easily proved that $\boldsymbol{X}^T\boldsymbol{Z} = 0$, i.e. $\boldsymbol{Z}$ is uncorrelated with $\boldsymbol{X}$. From this perspective, the corrected design matrix $\boldsymbol{Z}$ and $\boldsymbol{X}$ creates two-dimensional orthogonal subspaces. Variations or updates of random effects on the $\boldsymbol{Z}$-subspace will not influence fixed effects on the $\boldsymbol{X}$-subspace due to this orthogonal projection. This transformation helps limiting the explanation scope of random effects, i.e. random effects explain only the variation of response that cannot be explained by fixed effects.

Random effects selection

Theoretically, we can perform a componentwise Bayesian estimation for all potential random effects to perform the selection, e.g. by comparing model improvements of all pairwise random effects. But this is obviously not an efficient method, since thousands of Gibbs samples need to be drawn for each variable in every iteration, and most of them are discarded after the winner of the comparison is obtained. Therefore, we assume that random effect variables are a subset of fixed effect variables, i.e. what we have mentioned many times above that only the variable that has fixed effect can have random effect. By this way it is enough to draw Gibbs samples for only the selected fixed effect variables, which is especially efficient for high-dimensional data. This assumption coincides also with practical experience, namely, it is hard to imagine a variable that affects the response only in a random manner without including some degree of fixed effect at all. To put it simply, a variable in a boosting iteration is said to have an informative random effect when it has an informative fixed effect and the model improvement of treating its random effect is greater than its fixed effect.

To make this selection mechanism possible, the covariance structure of random effects needs to be reconstructed at the sampling step. Based on the random effects assumption, as long as the best-fitting fixed effect X_{k^*} in iteration s has not a random effect, a temporary or potential covariance matrix $\boldsymbol{Q}_{\text{pot}}^{[s]}$ shall be constructed as

$$\boldsymbol{Q}_{\text{pot}}^{[s]} = \text{diag}(\boldsymbol{Q}^{[s-1]}, 1),$$

with 1 as the initialized starting value. The change of the covariance structure affects also other relevant elements, for example the hyperparameter $\boldsymbol{\Lambda}_{0,\text{pot}}^{[s]}$. The design matrix $\boldsymbol{Z}_{\text{pot}}^{[s]}$ should also correspond to the new structure. Estimates, especially $\gamma_{\text{mode, pot}}^{[s]}$ (posterior mode), based on the new structure enables us to get access to the model improvement benefiting from the random effects part of X_{k^*}, i.e.

$$\text{MSE}_{k^*,\text{random}} = \frac{1}{n}\sum_{i=1}^{n}\left(u_i - \boldsymbol{Z}_{ik^*,\text{pot}}^{[s]}\hat{\gamma}_{k^*,\text{mode, pot}}^{[s]}\right)^2, \tag{3.9}$$

where $\boldsymbol{Z}_{k^*,\text{pot}}^{[s]}$ denotes the submatrix of $\boldsymbol{Z}_{\text{pot}}^{[s]}$ accounting for the k^*-th covariate.

Decisions on the selection of random effects can thus be made by comparing the mean squared error of the fixed effect part in (3.5) and the random effect part in (3.9) regarding X_{k^*}. If model improvement of the latter is greater than that of the former, the structures of potential covariance as well as other elements shall be held, otherwise they ought to be reset to the previous status.

Two reasons ensure the fairness of the comparison between the update in the fixed effects and the random effects. The first one lies in the underlying concept of boosting, i.e. update the parameter of interest yielding the largest improvement. The marginal influence of fixed effects decrease with increasing iterations, and upto some threshold random effects will be considered by the model. In other words, as long as the random effects are really informative, they will be selected into the final model sooner or later. The second one is the algorithm specification. On one hand, the subset assumption of random effects guarantees a variable's random effect cannot be selected earlier than its fixed effect. This reduces the risk of including too many noise random effects, since they are not shrunken and have large influence at early iterations and this indeed increases the chance of random effects being selected at early stage. But on the other hand, our algorithm specification makes random effects explain only the part that cannot be explained by fixed effects as discussed in the bias correction step. This means the information at early stage that is explained by random effects will be reexplained by fixed effects at later iterations.

Theoretically, if X_{k^*} is not additionally recognized to have a random effect, the MCMC simulation process should be conducted again with the old covariance structure to get more precise estimations. But we can still keep the samples excluding only X_{k^*} relevant values practically, because a larger MSE_{k^*} usually implies an uncorrelated non-informative random effect X_{k^*}, so ignoring the X_{k^*} relevant samples will have little effect on the estimation. And regarding efficiency, a second MCMC procedure adds a huge computational burden. Therefore, in our proposed algorithm the estimates sampled from the potential structures are updated even if the random effect of X_{k^*} is not selected.

3.1.3 Stopping criterion

Classical model selection techniques are not so useful for the proposed framework, since on the one hand, resampling methods like cross-validation (Allen, 1974; Stone, 1974, 1977) are difficult to be applied to mixed models because random effects are subjects-level effects making estimates from the training data useless for the validation set. On the other hand, information criteria like the cAIC fail due to the usage of MCMC simulation since estimation based on the drawn samples is naturally random. Thus, the global minimum of the cAIC series (i.e. a sequence of cAIC values, where each value represents the evaluation of the model in the corresponding boosting iteration) is subject to stochasticity and no longer reliable. Consequently, we suggest to use the

Algorithm 5 Probing for mixed models

```
1: Expand the data set $\boldsymbol{X}$ by creating randomly shuffled images $\tilde{X}_k$ for each of the $k = 1, \dots, p$ variables $X_k$ such that
       $\tilde{X}_k \in S_{X_k},$
   where $S_{X_k}$ denotes the symmetric group that contains all $n!$ possible permutations of $X_k$. Note that cluster-constant variables permute at the subject level.
2: Initialize a boosting model on the inflated data set
       $\bar{\boldsymbol{X}} = [X_1, \cdots, X_p, \tilde{X}_1, \cdots, \tilde{X}_p]$
   for the fixed effects estimation and start iterations with $s = 0$.
3: Stop if the first $\tilde{X}_j$ is selected.
4: Return only the variables selected from the original data set $\boldsymbol{X}$.
```

probing technique (Thomas et al., 2017).

The main idea of probing is adding artificial non-informative variables (usually the permutation of all observed variables since the marginal distribution is preserved) to the data to benefit from the presence of variables that are known to be independent from the outcome. It is straightforward to implement probing to the componentwise gradient boosting since boosting algorithms update a variable which yields the largest improvement, and selecting a artificial variable essentially implies the best possible improvement relies on information that is known to be unrelated to the outcome, i.e. it is overfitted at this stage.

For convenience, we have made some amendments to the algorithm proposed by Thomas et al. (2017) to make it suitable for mixed models, see algorithm 5. Note that for cluster-constant variables, their permutation shall be conducted at the subject-level. In addition, there is no need to extend the probing concept to random effects, since it will add a substantial computing burden while benefiting little from the precision. According to the mechanism of probing, the algorithm only stops when one of the artificial non-informative variables is selected. This means that as long as the model has adequately explained the information in the data, further estimates will inevitably select non-informative variables, be it fixed effect or random effect. The stopping iteration may differ when applying probing to random effects, but the difference can be neglected. From the computing concerns, we therefore consider probing only for fixed effects.

3.2 Simulation

In the following, three simulation studies are shown to demonstrate the performance of `BayesBoost`. The first one compares the estimation accuracy for both random intercept and random slope model between `BayesBoost` and the enhanced gradient boosting algorithm `grbLMM` Griesbach et al. (2021a). The second one highlights the random effects selection of `BayesBoost` and explores its uncertainty estimation feature. The performance of uncertainty estimation is compared with `BayesX` (Belitz et al., 2022) in the last simulation.

3.2.1 Estimation accuracy

The `grbLMM` algorithm shows the latest research results in applying gradient boosting technique to linear mixed models, but the covariance structure of the random effects must be specified in advance due to lacking an option for random effects selection. The `BayesBoost` algorithm can mimick this behaviour by preserving the space of limiting the maximal number of random effects the final model contains or just giving a pre-defined covariance structure. These can be obtained by skipping step 7 in algorithm 4 if the maximal number of random effects is exceeded, or simply replacing the step with a pre-defined covariance structure (and skip the random effects selection, i.e. comparison of MSEs), and the latter is what we have done in this simulation to conduct a fair comparison.

We use the same setup as in Griesbach et al. (2021a): for individuals $i = 1, \cdots, 50$ and their replicates $j = 1, \cdots, 10$ and thus the total observations $n = 500$, the response in the random intercept model is drawn from

$$\boldsymbol{y}_i = 1 + 2\boldsymbol{x}_{i1} + 4\boldsymbol{x}_{i2} + 3\boldsymbol{x}_{i3} + 5\boldsymbol{x}_{i4} + \boldsymbol{\gamma}_{i0} + \boldsymbol{\varepsilon}_i,$$

with $\boldsymbol{\gamma}_0 \sim \mathrm{N}(0, \tau^2)$, and in the random slope model is drawn from

$$\boldsymbol{y}_i = 1 + 2\boldsymbol{x}_{i1} + 4\boldsymbol{x}_{i2} + 3\boldsymbol{x}_{i3} + 5\boldsymbol{x}_{i4} + \boldsymbol{\gamma}_{i0} + \boldsymbol{\gamma}_{i1}\boldsymbol{x}_{i3} + \boldsymbol{\gamma}_{i2}\boldsymbol{x}_{i4} + \boldsymbol{\varepsilon}_i, \tag{3.10}$$

with $(\boldsymbol{\gamma}_{i0}, \boldsymbol{\gamma}_{i1}, \boldsymbol{\gamma}_{i2}) \sim \mathrm{N}(\mathbf{0}, \boldsymbol{Q})$ where

$$\boldsymbol{Q} = \begin{pmatrix} \tau^2 & \tau^* & \tau^* \\ \tau^* & \tau^2 & \tau^* \\ \tau^* & \tau^* & \tau^2 \end{pmatrix}.$$

The simulations are evaluated for five different cases $p \in \{10, 25, 50, 100, 500\}$ ranging from low to high dimensions. In both models random variables $\boldsymbol{x}_i, i = 1, \dots, p$ are standard normal distributed and only the first four variables are informative, specifically, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are cluster-constant covariates. Moreover, $\boldsymbol{\varepsilon} \sim \mathrm{N}(0, \sigma^2)$ with $\sigma = 0.4$ and $\tau \in \{0.4, 0.8, 1.6\}$. In the random slope model τ^* is chosen such that $\mathrm{cor}(\boldsymbol{\gamma}_{ic}, \boldsymbol{\gamma}_{id}) = 0.6$ for all $c, d = 1, 2, 3$ holds.

The estimation accuracy is evaluated by the mean squared errors

$$\mathrm{MSE}_{\boldsymbol{\theta}} := \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|^2 \quad \text{and} \quad \mathrm{MSE}_{\sigma^2} := \left(\sigma^2 - \hat{\sigma}^2\right)^2$$

with $\boldsymbol{\theta} \in \{\boldsymbol{\beta}, \boldsymbol{\gamma}\}$. For random intercept model, the variance τ^2 is evaluated with $\mathrm{MSE}_{\tau^2} := \left(\tau^2 - \hat{\tau}^2\right)^2$, and in case of the random slope model, the covariance matrix $\boldsymbol{Q}$ is measured by $\mathrm{MSE}_{\boldsymbol{Q}} = \|\boldsymbol{Q} - \hat{\boldsymbol{Q}}\|_F$ where $\| \cdot \|$ denotes the Frobenius norm. Performance of variable selection is evaluated by calculating the false positive rate (FP). False negatives do not occur in both methods and hence are omitted.

Table 3.1 and 3.2 summarizes the performance of 100 simulation runs for random intercept and random slope model respectively. The stopping iteration of `grbLMM` are determined by 10-fold cross-validation, and that of `BayesBoost` are by probing. The MCMC samples T drawn at each iteration in `BayesBoost` are set to 1000 and the step-length ν in both methods is set to 0.1.

Generally, except for the variance component, `grbLMM` slightly outperforms `BayesBoost` in all mean squared error measures for both random intercept and random slope models, but the differences are neglectable. However, the false positive rates of `grbLMM` are obviously worse than the ones produced by `BayesBoost` in all cases. This coincides with the outperformance of `grbLMM` in the MSE comparison, as a model with more false positives usually overfits and therefore has better estimation accuracy.

In turn, one of the big improvements of `BayesBoost` is the drastically reduction of

Table 3.1 Mean value of 100 simulation runs with respect to each model evaluation metric between `grbLMM` and `BayesBoost` in the random intercept setup.

τ	p	`grbLMM`					`BayesBoost`				
		MSE$_{\boldsymbol{\beta}}$	MSE$_{\boldsymbol{\tau}^2}$	MSE$_{\sigma^2}$	MSE$_{\gamma}$	FP	MSE$_{\boldsymbol{\beta}}$	MSE$_{\boldsymbol{\tau}^2}$	MSE$_{\sigma^2}$	MSE$_{\gamma}$	FP
0.4	10	0.013	0.001	<.001	1.192	0.48	0.017	0.001	<.001	1.223	0.11
	25	0.014	0.001	<.001	1.201	0.31	0.020	0.001	<.001	1.240	0.05
	50	0.015	0.001	0.001	1.194	0.20	0.021	0.001	<.001	1.275	0.03
	100	0.019	0.001	0.001	1.278	0.14	0.022	0.001	<.001	1.267	0.01
	500	0.021	0.001	0.001	1.241	0.04	0.029	0.002	<.001	1.262	0.02
	10000	0.027	0.002	0.001	1.278	<.01	0.013	0.002	<.001	1.181	<.01
0.8	10	0.043	0.014	<.001	2.570	0.49	0.048	0.018	<.001	2.770	0.10
	25	0.043	0.014	<.001	2.528	0.35	0.053	0.016	<.001	2.912	0.05
	50	0.050	0.012	0.001	2.759	0.24	0.053	0.016	<.001	2.806	0.02
	100	0.050	0.015	0.001	2.701	0.16	0.057	0.019	<.001	2.850	0.01
	500	0.057	0.015	0.001	2.837	0.05	0.064	0.017	<.001	3.031	<.01
	10000	0.062	0.017	0.002	2.757	<.01	0.070	0.015	0.001	2.695	<.01
1.6	10	0.155	0.230	<.001	7.850	0.47	0.178	0.304	<.001	8.989	0.11
	25	0.178	0.195	<.001	8.710	0.34	0.178	0.267	<.001	8.790	0.04
	50	0.176	0.259	0.001	8.413	0.29	0.185	0.315	<.001	9.066	0.02
	100	0.174	0.238	<.001	8.416	0.14	0.187	0.396	<.001	8.782	0.01
	500	0.166	0.255	0.001	7.823	0.05	0.186	0.265	<.001	8.486	<.01
	10000	0.179	0.304	0.003	7.826	0.01	0.178	0.313	0.001	7.823	<.01

false positives at a very low cost of accuracy. To some extent the effect can however be contributed to probing. As introduced above, probing determines the stopping iteration via the addition of non-informative variables, it is therefore sensitive to the false positives. Consequently, it delivers a lower number of false positives in contrast to common tuning procedures. For a more detailed comparison between probing and CV, please refer to Thomas et al. (2017). A potential drawback of probing is the stochasticity of permutations, there is thus no deterministic stopping iteration. Rerunning the algorithm with different random seeds can help to stabilize results, but there is no evidence proving resampling methods can help further reducing the false positives.

The simulation as conducted by Griesbach et al. (2021a) has the disadvantage of only covering one type of noise-to-signal ratio, we hence want to give a few details for a broader coverage of this matter. We alter the picture with the same settings corrected for the noise-to-signal ratio (NSR). The coefficients β_c are transformed according to

$$\beta_c = \beta\sqrt{(r\beta^T\Sigma\beta)^{-1}},$$

where $\beta = (2, 4, 3, 5)$ as the settings above and the NSR $r \in \{0.2, 0.5, 1\}$. The covariance matrix Σ is a Toeplitz matrix with $\Sigma_{ij}\rho^{|i-j|}$ for all $1 < i, j < p$, where $\rho = 0.9$.

Table 3.3 shows the averaged false positive rate (FPR) and false negative rate (FNR) among 100 simulation runs for both `grbLMM` and our proposed method. Unlike

Table 3.2 Mean value of 100 simulation runs with respect to each model evaluation metric between `grbLMM` and `BayesBoost` in the random slope setup.

τ	p	`grbLMM`					`BayesBoost`				
		MSE$_{\beta}$	MSE$_{Q}$	MSE$_{\sigma^2}$	MSE$_{\gamma}$	FP	MSE$_{\beta}$	MSE$_{Q}$	MSE$_{\sigma^2}$	MSE$_{\gamma}$	FP
0.4	10	0.020	0.013	0.002	4.482	0.46	0.026	0.010	<.001	3.529	0.10
	25	0.022	0.012	0.003	4.530	0.27	0.029	0.010	<.001	3.545	0.04
	50	0.023	0.012	0.003	4.551	0.18	0.030	0.011	<.001	3.588	0.02
	100	0.025	0.012	0.003	4.536	0.10	0.031	0.011	<.001	3.644	0.01
	500	0.027	0.011	0.003	4.453	0.03	0.040	0.010	<.001	3.660	<.01
	10000	0.032	0.012	0.005	4.548	<.01	0.054	0.009	0.001	3.952	<.01
0.8	10	0.072	0.124	0.002	6.923	0.44	0.083	0.133	<.001	6.332	0.11
	25	0.073	0.121	0.003	7.015	0.28	0.087	0.136	<.001	6.482	0.04
	50	0.074	0.119	0.003	6.956	0.17	0.087	0.134	<.001	6.620	0.02
	100	0.078	0.094	0.003	7.060	0.11	0.085	0.118	<.001	6.635	0.01
	500	0.082	0.124	0.003	6.953	0.04	0.100	0.139	<.001	6.832	<.01
	10000	0.083	0.130	0.004	6.804	<.01	0.090	0.124	0.001	6.157	<.01
1.6	10	0.280	1.829	0.002	16.970	0.41	0.321	2.053	<.001	15.669	0.09
	25	0.277	1.808	0.002	16.605	0.29	0.316	2.007	<.001	15.940	0.04
	50	0.294	1.435	0.002	17.124	0.19	0.302	1.796	<.001	15.950	0.02
	100	0.299	1.852	0.003	16.682	0.14	0.310	1.931	<.001	15.006	0.01
	500	0.320	1.804	0.003	17.658	0.04	0.361	1.773	<.001	17.280	<.01
	10000	0.281	1.904	0.005	16.380	<.01	0.339	1.788	0.001	15.705	<.01

in section 3.2.1 we can see the occurrence of false negatives with the help of NSR in this example. The `BayesBoost` approach includes relatively less ineffective variables than `grbLMM` at the cost of the exclusion of more effective ones. But generally speaking, for both methods, the FPR and FNR are on the low levels. Hence, the absence of false negatives in section 3.2.1 is mainly due to the strong signal of truly informative variables. Note that the equivalence of FPR for all NSRs actually comes by chance, since the estimated coefficients differ slightly from each other, though they are not shown here, and the FNR varies under different NSRs.

Table 3.3 False positive rate (FPR) and false negative rate (FNR) under different noise-to-signal ratio (NSR). The outcomes average 100 simulation runs with the total number of covariates $p = 100$ and $\tau = 0.8$.

	`grbLMM`		`BayesBoost`	
NSR	FPR	FNR	FPR	FNR
0.2	0.096	<.001	0.011	<.001
0.5	0.077	0.001	0.011	0.022
1	0.066	0.001	0.011	0.065

In addition, we would like to make some comments on how to deal with random effects for the prediction of new data. In mixed models, one needs to decide whether predictions should be based on the marginal distribution of the response or on the distribution that is conditional on the modes of the random effects Bates et al. (2015). Therefore, setting random effects to zero for new observations is one possibility, another one would be making predictions conditional on the modes of all the random effects.

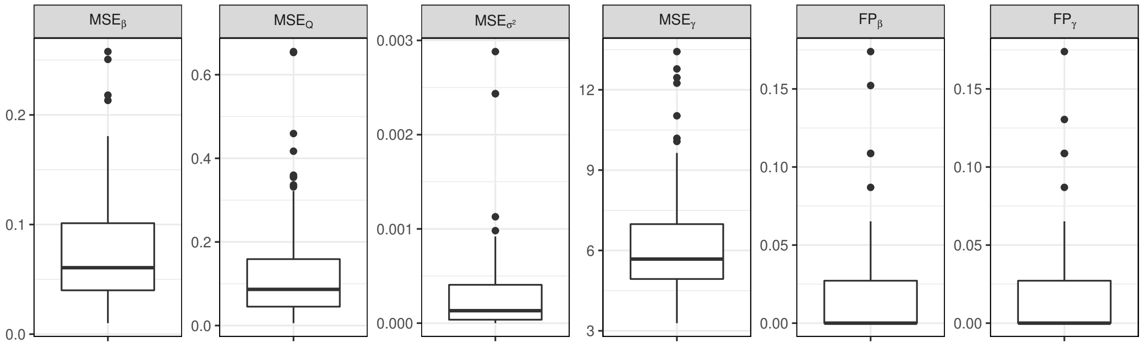


Figure 3.1 Boxplot of each model evaluation metric estimated by `BayesBoost` summarizing the outcomes of 100 simulation runs for the random slope setup with $\tau = 0.8$ and $p = 50$.

3.2.2 Random effects selection

To explore the performance of random effects selection in `BayesBoost` as well as other features, we use the same simulation settings as in equation (3.10) and select only one typical setup with $\tau = 0.8$ and $p = 50$, but let all covariates (except for the cluster-constant covariates) be participants for random effects. In other words, we do not specify the covariance structure of random effects in advance, but let the algorithm choose them automatically.

Figure 3.1 illustrates the distribution of each model evaluation metric defined above over 100 simulation runs and all of the models are estimated by `BayesBoost`. Due to the free structure of random effects, a false positive rate to $\boldsymbol{\gamma}$ is also shown in this figure. In addition, since there is no occurrence of false negatives for both fixed effects and random effects, they are hence omitted.

Overall speaking, there is almost no obvious difference between this outcome and the corresponding row in Table 3.2. In particular, the false positive rate for random effects (FP$_{\boldsymbol{\gamma}}$) is zero in more than half of the simulations, and its third quantile of 0.025 indicates that the majority of the runs select at most one non-informative random effect. Therefore, from this comparison we can find that even though the random effects are not given in advance, the `BayesBoost` algorithm can still capture the structures effectively and give competitive outcomes.

To explore the characteristics of `BayesBoost`, we take a typical simulation from the 100 runs as an example. In this example, none of the non-informative variables is selected as fixed or random effect and all informative ones are included in the the final

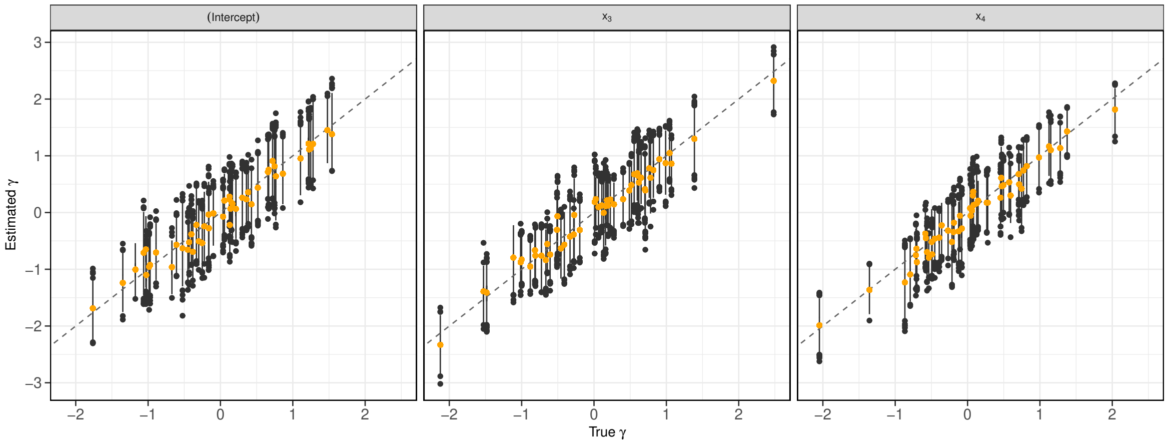


Figure 3.2 Distributions of the MCMC samples with respect to all individuals at the stopping iteration. Black lines summarize the distribution of the samples with outliers marked with the black dot. Orange points denotes the sample modes (posterior modes) of each individual. The dashed grey line marks the location where the estimates coincides with the truth.

model.

Figure 3.2 illustrates the distribution of MCMC samples for each random effect at the stopping iteration. According to the setup, the size of the sample $T = 1000$ and they are summarized by black lines with respect to every individual (and outliers marked with black points). The orange points represent the (posterior) modes of the samples and serve as the estimates of random effects $\hat{\gamma}$. The theoretical ideal estimation is given through the dashed grey reference slash.

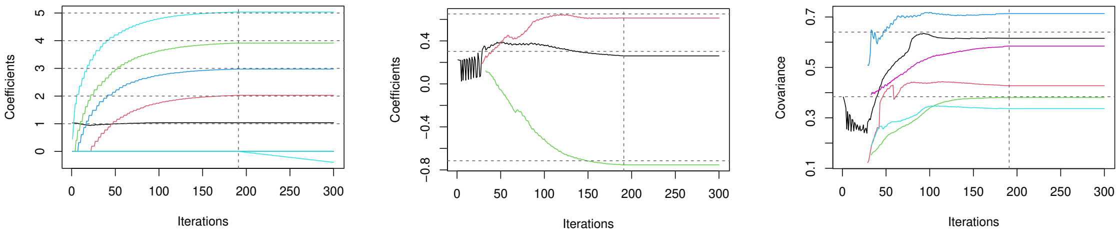
Besides again illustrating the fit of the $\hat{\gamma}$ point estimation, the more important feature lies in that `BayesBoost` is able to produce the uncertainty estimation for random effects. Given the sample distribution, parameter estimation using boosting technique is no longer simply a point estimation, but we can further tell to which extent we accept the estimates. Hypothesis test, credible intervals as well as other Bayesian statistics of interest for the random effects (and also for their covariance matrix $\boldsymbol{Q}$) can thus be established based on the samples.

It is possible to get access to similar uncertainty estimation for random effects from conventional boosting methods, namely with the help of permutation or bootstrap. But these methods suffer from the bias induced by shrinkage of boosting. `BayesBoost`, however, gets rid of the shrinking estimation method but estimates random effects all at once in each iteration. This makes the uncertainty obtained from `BayesBoost` more reliable.

Figure 3.3 shows the estimated coefficient paths of each covariates in each boosting iteration for fixed effects $\hat{\boldsymbol{\beta}}$, the random effects $\hat{\gamma}_i$ (posterior mode) of an individual i and the estimated variance-covariance matrix $\hat{\boldsymbol{Q}}$. The stopping iteration is marked with the vertical black dashed line. As discussed above, `BayesBoost` estimates random effects not through the sum up of learning pieces but altogether as a whole. Hence shrinkage is only applied to the estimation of fixed effects but not the estimation of random effects. The other important boosting feature, variable selection, is shared by both fixed and random effects. In this example, up to the stopping iteration, all informative fixed and random effects are included into the final model with well fitted coefficients.

If we take a closer look at figure 3.3b, we will find that the coefficient paths can be roughly divided into three periods. The first period from the beginning lasts to about 30 iteration. Estimates in this period oscillate heavily mainly due to the burn-in of MCMC simulations. The second period lasts up to 191 iteration, which is also the stopping iteration. In this period, along with the convergence of fixed effects, random effects converge relatively smoothly to the true values. The remaining iterations can be grouped to the third period. Fixed effects for this period have been overfitted, while estimates for random effects fluctuate around their converged values. The degree of oscillation depends on the Gibbs samples, and the fewer the samples, the greater the oscillation of the curve wave.

As mentioned above that the seemingly "convergence" of random effects in the second period cannot be interpreted as the shrinking estimation as fixed effects. Since random effects in each `BayesBoost` iteration capture the residual information as much as possible, each model during this period is already a mature model. This can be observed more clearly from the similar graphic for the covariance matrix $\hat{\boldsymbol{Q}}$ in figure 3.3c. The covariance between random effects changing little from about the 100 iteration to the stopping iteration indicates the covariance matrix have already been in good state. However, the coefficients of random effects in figure 3.3b still shows the convergence behavior during this period. This phenomenon is actually the consequence of the changes of fixed effects in this period. As discussed above, random effects should only explain the response that cannot be explained by the fixed effects. This view takes the form in practice that most of the data information is explained by random effects in



(a) Fixed effects $\hat{\boldsymbol{\beta}}$ **(b)** Random effects $\hat{\gamma}_i$ **(c)** Covariance matrix $\hat{\boldsymbol{Q}}$

Figure 3.3 The estimated coefficients of fixed and random terms as well as the estimated covariance of random effects in each boosting iteration with the stopping iteration marked with the dashed vertical line and the true values marked with the vertical dashed lines. Plot (a) shows the estimates of fixed effects $\hat{\boldsymbol{\beta}}$. Plot (b) displays the estimated random effects $\hat{\gamma}_i$ for an individual i, where each curve in this plot is drawn by the the sample modes. The black curve represents the random intercept, the red and green curves represent the estimates for random slope $\boldsymbol{x}_4$ and $\boldsymbol{x}_3$ respectively. Plot (c) shows the estimates of covariance $\hat{\boldsymbol{Q}}$ in each iteration. According to the model specification, the true variance (diagonal of $\boldsymbol{Q}$) for each random effect is 0.64, and the true covariance between random effects (off-diagonal) is 0.384. Both true values are marked with the dashed horizontal grey line. Counting from top to bottom, the first three lines at the stopping iteration are the variance of random effects, while the last three lines are the covariance between random effects.

early iterations, since the fixed effects due to shrinkage are quite small at this stage, and as the fixed effects increase in later iterations, those information explained by random effects earlier turns to be explained by fixed effects. This process thus makes estimates of random effects forming a seemingly convergence behavior.

In addition, figure 3.3c is also drawn by the elementwise posterior modes of the covariance samples. That means the Bayesian analysis of random effects also applies to their covariance structure.

3.2.3 Performance of uncertainty estimation

The last simulation is to demonstrate the uncertainty estimation performance of `BayesBoost` by a comparison to `BayesX` (Belitz et al., 2022). `BayesX` is a popular and well-established tool for analyzing Bayesian structured additive regression models based on MCMC simulation techniques such as generalized additive models (GAM) and generalized additive mixed models (GAMM) that are important to this chapter.

Since the variable selection logic is different between `BayesX` (penalized likelihood based) and `BayesBoost` (boosting based), for a relatively fair comparison, we specify full

Bayesian inference in `BayesX`, which is the same as `BayesBoost`. The simulated data is the same as equation (3.10) with $\tau = 0.8$, but without noise variables, since the selection performance has been discussed and illustrated above. Therefore, a random effects structure is given in advance. Other specifications include the sample size $T = 1000$ and step-length $\nu = 0.1$. The results for `BayesX` are based on 12000 iterations and using every 10th sampled parameter for estimation after the burn-in period of 2000 iterations. Both algorithms are applied to 100 simulation runs.

Figure 3.4 illustrates typical interval estimates of random effects for all individuals from one the 100 runs for `BayesBoost` and `BayesX`. It can be observed that the estimated intervals of `BayesBoost` are not very different from those of `BayesX`, since most of the true effects are covered by the 95% credible interval. Only few true effects lie out of the estimated intervals, for example the random intercept of the last individual. Overall speaking, the distinct differences in the estimates for different individuals and the fact that the estimated intervals generally cover the true effect indicate that both algorithm are effective for estimating random effects, especially the effectiveness of `BayesBoost`.

The effectiveness can be observed more clearly from figure 3.5, which illustrates the coverage probabilities of the 80%- and 95%-intervals for all 100 simulation runs. For example, suppose the 80% credible interval of `BayesBoost` in figure 3.4 covers the true effects of 42 out of all 50 individuals, then it results in a value of 0.84 (= 42/50). Summarizing all the coverage probabilities in 100 simulation runs we get average values as marked in each subfigure.

Different from figure 3.4, which shows only a graphical similarity of the two methods in a single simulation run, figure 3.5 provides more complex quantitative information that summarizing the performance in more simulation runs. In general, both `BayesBoost` and `BayesX` share the same graphical pattern, i.e. both approaches have low coverage probabilities in some simulation runs and high probabilities in others. This pattern clearly shows the uncertainty estimation of the proposed algorithm is on the same level as `BayesX`. From the quantitative perspective, the fact that the coverage rate of 80%- and 95%-intervals are over 0.8 and 0.95, respectively, indicating that both algorithms can estimate random effects well. The shows also the good performance of `BayesBoost` in uncertainty estimation.

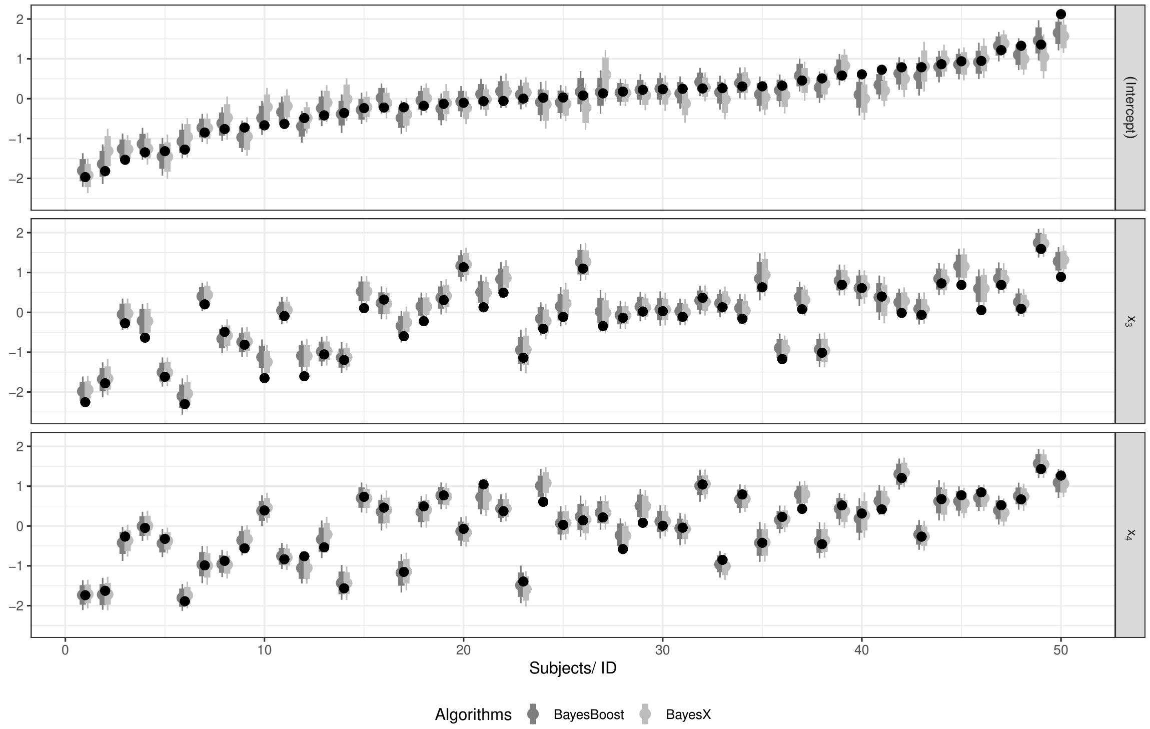


Figure 3.4 Interval estimates of random effects for each individual between `BayesBoost` (dark grey) and `BayesX` (light grey) sorted according to the ascending order of random intercept. The interval estimate for each individual displays with two lines, where the thick line covers the 80% of all MCMC samples (i.e. 10% and 90% quantiles), and the thin line covers 95% of the samples (i.e. 2.5% and 97.5% quantiles). The median is marked with grey points. The black dot in the middle of two grey lines for each individual indicates the true effect.

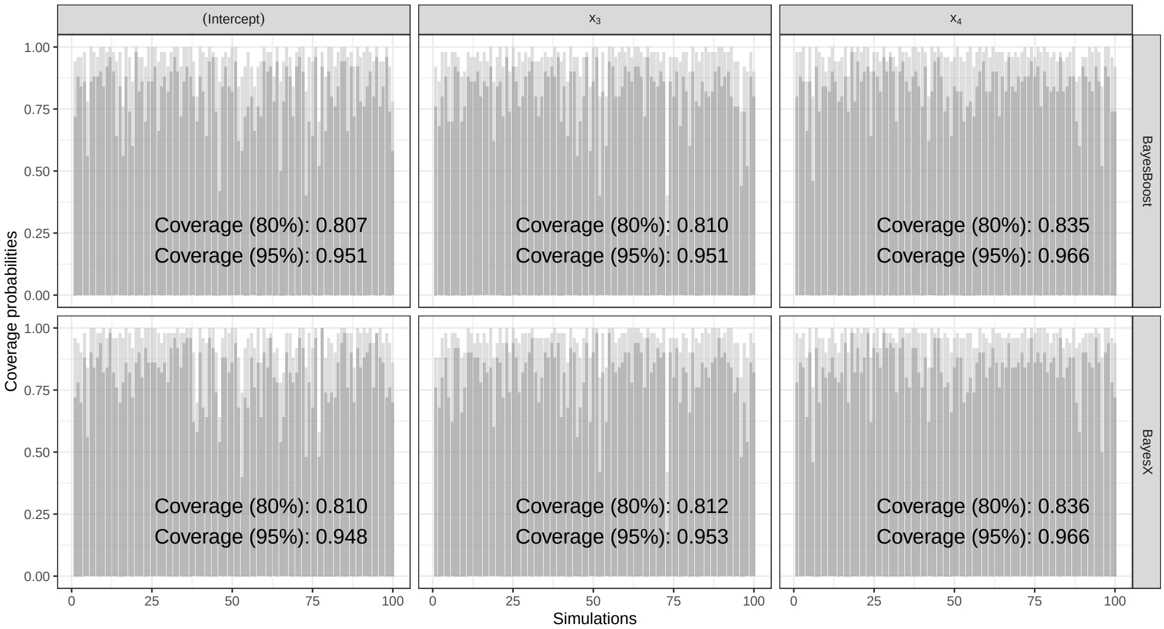


Figure 3.5 Coverage probabilities of the 80%- and 95%-intervals of each random effect in 100 simulation runs for both by `BayesBoost` and `BayesX`. For each run, the coverage probability summarizes the percentage of true effects covered by the corresponding interval. The dark and light grey bars in each simulation indicate the 80%- and 95%-interval respectively. The overall coverage rate among all 100 runs are labeled with the corresponding values.

3.3 Application

We apply our proposed algorithm to a real data example, riboflavin (Vitamin B_2) production by Bacillus subtilis which is provided by DSM (Switzerland) and was first published in Schelldorfer et al. (2011). The dataset has $m = 28$ specimens measured at two to six time points (i.e. $n_i \in \{2, \dots, 6\}$) and a total of $n = 111$ observations. The response variable is the logarithm of the riboflavin production rate and the gene expression levels are measured by $p = 4088$ covariates (genes). Therefore, this data set calls for a strong variable selection tool, which is adapted to mixed models.

Due to the stochasticity of `BayesBoost` for both MCMC sampling and probing, there is no deterministic solution to the data analysis if we rerun the algorithm multiple times without a fixed random seed. We thus perform the algorithm 100 times with different random MCMC seeds to stabilize outcomes. The sample size was chosen to be $T = 1000$ and the step-length $\nu = 0.1$. The random structure is not given in advance, such that algorithm select random effects automatically.

Previous findings (Meinshausen et al., 2009; Lin et al., 2020) indicate *YXLD-at* as a significant fixed effect in riboflavin production. The impact of *YXLD-at* is also found by other authors (Javanmard and Montanari, 2014; Meinshausen et al., 2009; Bühlmann et al., 2014), who use a homogeneous version of the riboflavin set with $n = 71$ but from the same source, which is available in the R package `hdi`. Note that due to the lack of possibility to select random effects in these studies, their outcomes are merely based on random intercept models.

Compared with the approaches in these studies, a key advantage of `BayesBoost` lies in the random effects selection feature, which enables us to apply the more general random slope model to the data, and let the algorithm decide what the final model looks like. As with previous findings, we also observe the important effect of *YXLD-at* on the riboflavin production, but this effect is not restricted to the role of fixed effect, it affects the response also in a random way. On one hand, *YXLD-at* is selected as a fixed effect in all 100 reruns, followed by *LCTE-at* and *ssuA-at* 39 times each. And on the other hand, all of the models in these 100 reruns also conclude *YXLD-at* as a random effect, followed by *PRIA-at* 20 times and *YVAK-at* 13 times. Though a significant test is not performed in our studies, we still have enough confidence to say that *YXLD-at* is an important effect if we average all the 100 outcomes.

One possible interpretation of this finding could be that the impact of *YXLD-at* is twofold: on one hand we find a strong deterministic influence of this gene, but there is also an subject-specific impact that has to be attributed to each individual. On the other hand, this can be either some other covariate we cannot measure, but which is correlated with *YXLD-at* or simply a different strength of the impact of *YXLD-at* itself.

To look at it from the statistical point of view, we refer to figure 3.6 which illustrates the distributions of the mode of random effect *YXLD-at* for each specimen over the 100 reruns. On one hand, modes, as well as credible intervals, show clear differences among specimens. This suggests that there are individual differences in the effects of the riboflavin production. On the other hand, it can be observed that the 100 outcomes have a relatively consistent conclusion on the estimates of the random effect *YXLD-at* since most of the estimates concentrate on their mode, in other words, their standard deviations (or variances) are small. This also strengthens our judgment that *YXLD-at* affects the response in an individually random way.

This chapter is to the best of our knowledge the first to propose that *YXLD-at* is

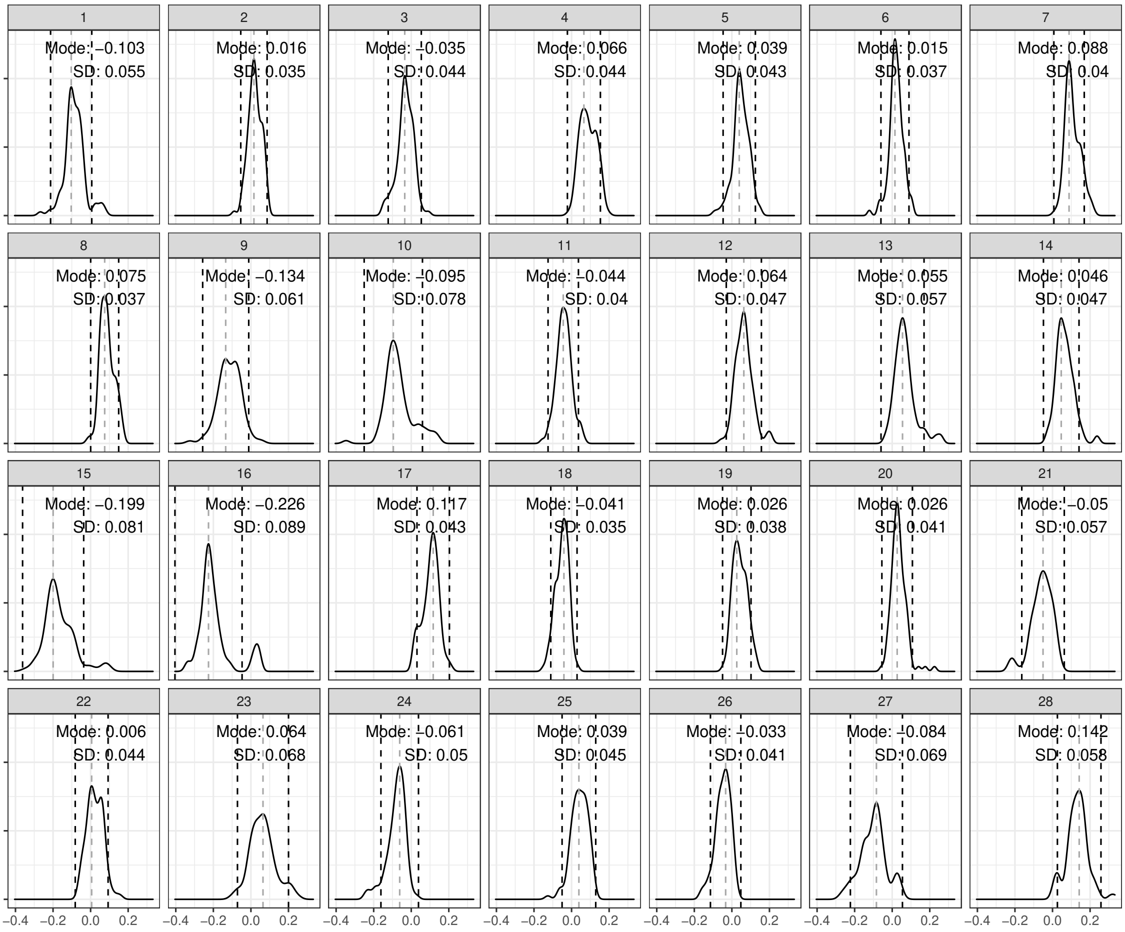


Figure 3.6 Densities of the mode of random effect *YXLD-at* for each specimen over the 100 outcomes. The mode of the density and the standard deviation is labeled with a value at the top right corner respectively. Modes are also marked with dashed grey lines and two standard deviations around modes are marked with dashed black lines.

not only a fixed effect but also a random effect in the riboflavin production.

3.4 Discussion and Outlook

One of the most important reasons for boosting techniques being widely used in statistics is due to its appealingly direct and effective variable selection feature. However, as the method originated from machine learning, it lacks straight forward ways to construct estimators for the precision of parameters such as variance or confidence intervals like other statistical approaches based on distributions. Thanks to the development in computer science, the Bayesian inference has grown immensely in the last decades and rendered possible an extreme amount of new types of models. But it very often fails to give precise and unambiguous guidelines for the selection of variables.

This chapter proposes a new inference method, `BayesBoost`, by integrating a Bayesian learner into gradient boosting to benefit from both worlds. On one hand, the new approach has preserved the variable selection feature of model-based boosting methods, so that the parameter estimation and both fixed and random effects selection can be performed simultaneously. On the other hand, variation of random effects is accessible through the `BayesBoost` estimation, which is not possible in the conventional boosting framework. The effectiveness of `BayesBoost` can be observed from simulation and empirical studies.

However, as a new attempt, it leaves also some open questions: Firstly, the concept of `BayesBoost` is applicable due to the specific form of mixed models, i.e. the natural separation of fixed and random effects. But it is still unknown whether this concept can be extended to other common statistical models, especially the generalized additive models (GAM) or structured additive regression models (STAR). Secondly, even for the linear mixed models, `BayesBoost` only fills in the uncertainty estimation blank of the random effects part left by the boosting framework, that of the fixed effects part still remains unsolved. Thirdly, a potential drawback of our proposed algorithms is the side effect of the shrinkage parameter ν on the selection preference of random effects. According to the selection mechanism (step 12 in algorithm 4), a variable is said to have random effect if the global improvement of it's random effect is greater than it's fixed effect, but the latter is affected by the shrinkage parameter ν. A preliminary decision thus have to be made, i.e. a smaller ν is suggested if one favors random effects, and vice versa. We set it here to 0.1, but tuning it with cross-validation or other possible methods needs to be further studied.

Extensions of this study could include the performance of hypothesis testing and the establishment of credible intervals. Tests about the effectiveness of the approach on the non-linear or spatial base-learners are also meaningful. Improving computational efficiency has never been an outdated topic in Bayesian statistics, and replacing computationally intensive MCMC simulations by much faster integrated nested Laplace approximation (INLA) seems to be a straightforward and effective way to accelerate the computing speed.

Since this chapter proposes only the preliminary Bayesian-based boosting concept, it focuses on the specific linear mixed model. The open questions discussed above, especially the problem of the generalization of the idea of Bayesian-based boosting to other common models will be discussed in the following Chapter 4.

Chapter 4

Bayesian-based Boosting for Quantifying Uncertainty in Structured Additive Regression

The boosting technique has been widely used in making inference for statistical models due to its stable variable selection feature and flexibility regarding the type of predictors. From the original boosting algorithm (Schapire, 1990; Freund, 1995), which aims to obtain a strong predictor by combining the solutions produced by iteratively applying simple weak classifiers, to Adaboost (Freund and Schapire, 1996, 1997), which has been hailed as the "best off-the-shelf classifier in the world" (Hastie et al., 2009), and later to gradient boosting (Friedman et al., 2000; Friedman, 2001), which adapts the concept of boosting to the field of statistical modeling, boosting has been implemented into almost all statistical topics over the last two decades.

One of the most successful variant of boosting in statistical learning is the componentwise gradient boosting (Bühlmann and Yu, 2003), which updates only one additive base-learner in each iteration. This simple but effective idea not only reduces the high-dimensional analysis to a simple regression problem (Bühlmann, 2006), but also provides the flexibility to estimate various types of base-learners in one additive regression model. The general estimation method used in gradient boosting is the least squares method. In low-dimensional settings, another typical inference method is the maximum likelihood. Based on this idea, Tutz and Binder (2006) propose likelihood-based boosting, in which the base-learners are directly estimated via optimizing the overall likelihood by using the additive predictor from the previous iteration as offset (Tutz and Binder, 2007; Groll and Tutz, 2012). Generally, likelihood-based boosting (including the componentwise likelihood-based boosting, which implements the componentwise concept) generates similar results to gradient boosting, and especially in the case of L_2 loss, likelihood-based boosting coincides with gradient boosting. However, in contrast to gradient boosting, approximate confidence intervals can be obtained by likelihood-based boosting (Tutz and Binder, 2006).

Other researches pay more attention to model generalization and regularization techniques. For example, boosting has already been implemented to generalized additive models (GAM) (Tutz and Binder, 2006; Schmid and Hothorn, 2008; Hofner et al., 2014; Hothorn et al., 2022), the more complex generalized additive models for location, scale and shape (GAMLSS) (Mayr et al., 2012; Thomas et al., 2018; Zhang et al., 2022b), the generalized additive mixed models (Groll and Tutz, 2012), and Cox models (Binder and Schumacher, 2008; Binder, 2013; De Bin, 2016). In addition to the inherent property of the componentwise concept in variable selection, the combination between boosting and ridge (Tutz and Binder, 2007) or lasso (Zhao and Yu, 2004) has also been investigated. Moreover, the selection performance has been enhanced and improved by using the stability selection approach (Meinshausen and Bühlmann, 2010; Thomas et al., 2018). For more details on the evolution of boosting, please review Mayr et al. (2014, 2017a).

Although more and more studies regarding statistical boosting have been published in recent decades, The majority of the published papers regarding boosting techniques focus on improving the estimation accuracy or in combination with various statistical models. Nevertheless, to the best of our knowledge, little literature has studied the fusion of boosting and Bayesian statistics, the latter of which, however, occupies half of modern statistics due to its unique philosophical perspective and also the computational advantages. In the few papers we found that contain both keywords (Bayesian and boosting), Elkan et al. (1997) propose a boosted naive Bayesian learner, which is equivalent to standard feedforward multilayer perceptrons. Similar models as well as their improvements can also be found in Bauer and Kohavi (1999); Ting and Zheng (1999), but due to their early publication, they are limited to the AdaBoost framework. Another paper (Nock and Sebban, 2001) proposes a so-called "Bayesian boosting theorem", which concernes the AdaBoost as well, aiming, however, at bounding the error of the boosting algorithm and increasing the convergence speed instead of combining the two (Bayesian and boosting) philosophies. The most relevant yet unpublished work by Lorbert et al. (2012), while still built on AdaBoost, performs approximate inference about the posterior distribution associated with latent variables or weights placed on the base classifiers. In addition, the quality of the learned classifier can be measured by the noise statistics of the classifier produced by the algorithm.

Even though Tutz and Binder (2006) point out the possibility of constructing approximate confidence intervals in likelihood-based boosting, this concept did not receive much attention until the publication of Rügamer and Greven (2020), which may be the first systematic study of uncertainty quantification in boosting. They propose inference for L_2-boosting in the special case of linear, grouped, and penalized additive models selected by L_2-boosting using the selective inference framework (Fithian et al., 2014; Tibshirani et al., 2016; Yang et al., 2016), a method that transfers classical statistical inference to algorithms with preceding selection of model terms. Compared to the previous ad-hoc solutions such as the permutation test (Mayr et al., 2017b), which is restricted to certain special cases, or the bootstrap (Brockhaus et al., 2015; Rügamer et al., 2018; Hepp et al., 2019), which does not lead to confidence intervals with proper coverage due to the bias induced by the shrinkage effect, the advantage of using a classical statistical method to quantify the uncertainty of boosting estimates is obvious. Yet, as possibly the first paper studying uncertainty in boosting, it focuses only on special additive models. Therefore, further work is still needed to cover a more general model family.

Instead of using this frequentist statistical approach to quantify uncertainty in boosting, we provide a Bayesian solution. In our previous studies (Zhang et al., 2022a), we introduced a method that integrates a Bayesian learner into the boosting framework for linear mixed models. However, the proposed algorithm was still preliminary, only providing uncertainty information for random effects. In this chapter, we extend this method by proposing a more general boosting framework based on Bayesian methodology, that integrates Bayesian penalized regression in the componentwise boosting framework. Compared to the previous work, the more general approach not only makes it possible to extract uncertainty information from the fixed effects, but also extends the model family to the generally structured additive regression (STAR) models, which is friendly to nonlinear and spatial base-learners.

In addition to richer technical features, this novel approach fills the gap in the field of applying Bayesian inference to boosting. In contrast to the dogmatic estimate of boosting, which delivers only an unquestionable point estimation, the proposed method not only benefits from the uncertainty and prior knowledge of Bayesian methods, but also maintains the useful features of boosting, for example, the intuitive variable selection procedure and the flexibility of various types of base-learners. Furthermore, this combination also provides a new way of thinking about the regularization research of classical Bayesian methods. We denote our proposed novel method as *Bayesian-based boosting* or `bboost` throughout this context.

This chapter is structured as follows: Section 4.1 describes briefly the basics of Bayesian penalized regression and componentwise gradient boosting and then proposes the details of the Bayesian-based boosting algorithm. In section 4.2, we compare our method with other commonly used methods through simulations of linear and non-linear scenarios, especially the performance in estimation accuracy, uncertainty, and variable selection. An empirical study, which analyzes the Munich rent index data including additional spatial variables, is presented in section 4.3. The study helps to demonstrate the effectiveness of the proposed method in STAR models. The final section 4.4 summarizes the chapter and discusses potential improvements of the proposed method and possibilities of other relevant further works.

4.1 Methods

Bayesian-based boosting is established on the basic structure of componentwise boosting, and instead of carrying out inference with least squares estimation in gradient boosting or maximum likelihood estimation in likelihood-based boosting, we conduct Bayesian inference for the unknown parameters in the new approach. In this section, we will first briefly introduce the concepts of Bayesian penalized regression and the idea of componentwise boosting, and then explain how to adapt them to make them suitable for constructing the new Bayesian-based boosting algorithm. Lastly, we will specifically discuss the mathematical relationship of the uncertainty in the new approach.

4.1.1 Model specification

Given the $(n \times p)$-dimensional matrix of covariates $\boldsymbol{X}$, the distribution of the response variable y belongs to an exponential family with mean $\mu = \mathbb{E}(y|\boldsymbol{X})$ linked to a linear predictor η by

$$\mu = h^{-1}(\eta),$$

and the general and flexible, structured additive predictor in STAR models (Fahrmeir et al., 2004; Brezger and Lang, 2006) is given by

$$\eta = f_1(\boldsymbol{z}_1) + \cdots + f_p(\boldsymbol{z}_p),$$

where various types of functions $f_j, j = 1, \ldots, p$ are defined on a generic vector of modeled covariates $\boldsymbol{z}_j$, for example, for linear cases, the function $f_j(\boldsymbol{z}_j) = \boldsymbol{x}^T\boldsymbol{\beta}$, where $\boldsymbol{x}$ is a subvector of $\boldsymbol{z}_j$; for non-linear cases, the function $f_j(\boldsymbol{z}_j) = f(x)$, and a spline function f is defined on the single element x of $\boldsymbol{z}_j$; for spatial cases, $f_j(\boldsymbol{z}_j) = f_{\text{spat}}(s)$, where s is a spatially correlated variable; for random intercepts with cluster index c, $f_j(\boldsymbol{z}_j) = \beta_c$, and for random slopes $f_j(\boldsymbol{z}_c) = x\beta_c$. Other types of effects such as two-dimensional surfaces or spatially varying effects are also covered by STAR models, for more details, please refer to Umlauf et al. (2015).

Due to the fact that all predictors can be written in terms of a linear combination of basis functions, the unified representation in matrix notation can be written as

$$\boldsymbol{f}_j = \boldsymbol{Z}_j\boldsymbol{\beta}_j,$$

where $\boldsymbol{f} = (f_j(\boldsymbol{z}_1), \ldots, f_j(\boldsymbol{z}_n))^T$. Then, the question simplifies to the generic model

$$\eta = \boldsymbol{Z}\boldsymbol{\beta},$$

where the design matrix $\boldsymbol{Z} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_p)$ and the corresponding coefficient $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_p)$. Therefore, for Gaussian distributed error term ϵ, the conditional distribution of the response is a linear model given by

$$y|\boldsymbol{\beta}, \sigma^2 \sim \mathrm{N}(\boldsymbol{Z}\boldsymbol{\beta}, \sigma^2\boldsymbol{I}),$$

where σ^2 denotes the variance of ϵ and $\boldsymbol{I}$ is the identity matrix.

4.1.2 Bayesian penalized regression

We use Bayesian penalized regression to estimate the unknown parameters since it has some similarities to the penalized least squares method. To obtain a Bayesian version of penalized regression, a particular prior needs to be specified, i.e.

$$\tilde{\boldsymbol{\beta}}|\tau^2 \sim \mathrm{N}(\boldsymbol{0}, \tau^2\boldsymbol{K}^{-1}),$$

where $\boldsymbol{\beta} = (\beta_0, \tilde{\boldsymbol{\beta}})^T$ and τ^2 denotes the variance parameter. The $\boldsymbol{K}$ is the penalty matrix, for example, $\boldsymbol{K} = \mathrm{diag}(0, 1, \ldots, 1)$ in case of Ridge/LASSO regression, and when considering the first-order difference penalty matrix for the smooth term, then $\boldsymbol{K} = \boldsymbol{D}_1^T\boldsymbol{D}_1$ and $\boldsymbol{D}_1$ is the first-order difference matrix. It can be proved that the posterior for $\boldsymbol{\beta}$ is a multivariate Gaussian distribution $\mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ with

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{\beta}} &= \left(\frac{1}{\sigma^2}\boldsymbol{Z}^T\boldsymbol{Z} + \frac{1}{\tau^2}\boldsymbol{K}\right)^{-1}, \\ \boldsymbol{\mu}_{\boldsymbol{\beta}} &= \frac{1}{\sigma^2}\boldsymbol{\Sigma}_{\boldsymbol{\beta}}\boldsymbol{Z}^T y.\end{aligned}$$

Maximizing this posterior with respect to $\boldsymbol{\beta}$ is equivalent to minimizing the penalized least squares criterion

$$\mathrm{PLS}(\lambda) = (y - \boldsymbol{Z}\boldsymbol{\beta})^T (y - \boldsymbol{Z}\boldsymbol{\beta}) + \lambda\tilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{\beta}},$$

with $\lambda = \sigma^2/\tau^2$. Similar to how the penalty term λ affects the estimation, the posterior mode is actually governed by τ^2, but in the reverse direction, i.e. the penalization is strong for small τ^2 and negligible for large values of τ^2.

Usually, we specify a conjugate prior or, more specifically, an inverse gamma prior with hyperparameters a and b for the variance parameter σ^2, i.e. $\sigma^2 \sim \mathrm{IG}(a, b)$. Thus, it can be proved that the full conditional distribution is again inverse gamma distributed

$\mathrm{IG}(\tilde{a}, \tilde{b})$ with

$$\begin{aligned}
\tilde{a} &= a + \frac{n}{2}, \\
\tilde{b} &= b + \frac{1}{2} \left(y - \boldsymbol{Z\beta}\right)^T \left(y - \boldsymbol{Z\beta}\right).
\end{aligned}$$

Unlike the general Bayesian penalized regression, in which an additional prior for the penalty term τ^2 is needed for its estimation, we regard τ^2 as a model hyperparameter controlling the degree of shrinkage of the estimation similar to the learning rate in boosting techniques. Depending on the practical situation, it can either be given a pre-defined fixed value or be tuned from the outside via, for example, cross-validation.

4.1.3 Componentwise boosting

In addition to the Bayesian foundations, we still need to modify the gradient boosting framework to establish a Bayesian-based boosting. We focus on the componentwise boosting since it is friendly to high-dimensional data, especially when the number of covariates is larger than the number of observations, which is a weakness of classical statistical methods, and therefore also of Bayesian statistics.

Generally, the fundamental of regression analysis using boosting techniques is the optimization problem, or more precisely, the minimization of the empirical risk

$$\arg\max_{\eta} \rho(y, \eta),$$

where ρ denotes a loss function. The most common loss function used in practice is the L_2 loss $\rho(y, \eta) = (y - \eta)^2$. Unlike the AdaBoost, which fits base-learners to re-weighted observations, the gradient boosting fits them to negative gradient vector $u^{[m]}$ of the loss function evaluated at the previous iteration

$$u^{[m]} = -\frac{\partial}{\partial \eta} \rho(y, \eta) \Big|_{\eta = \hat{\eta}^{[m-1]}}.$$

In case of the L_2 loss, the negative gradient vector simplifies to $(y - \hat{\eta})$. Therefore, the negative gradient vector is also called *pseudo-residuals* in some literature. To address the high-dimensional problem, Bühlmann and Yu (2003) proposed the *componentwise* idea, which fits every base-learner $h_j, j = 1, \ldots, p$ to $u^{[m]}$. Typically, each base-learner

h_j corresponds to one covariable, and only the best-performing base-learner, i.e. the one that yields the largest model improvement or the smallest square error loss, is added to the previous predictor and serves as the estimate for the current iteration.

In contrast to gradient boosting, where estimation is usually performed by the least square criterion, the estimation in likelihood-based boosting, just as its name suggests, is obtained by maximizing a likelihood, using the predictor from the previous iteration as an offset. In the special case of L_2 loss, likelihood-based boosting coincides with gradient boosting. Similarly, to make it suitable for high-dimensional data, likelihood-based boosting can also follow the componentwise routine and update only the best-performing component, which yields the largest log-likelihood in each iteration.

4.1.4 Bayesian-based boosting

Similar to the idea of likelihood-based boosting, that is the parameter of interest is estimated through likelihood, inference can also be made in the Bayesian way. Therefore, we build a Bayesian-based boosting framework by recursively estimating the pseudo-residuals (using L_2 loss) in each iteration with the Bayesian penalized regression. Algorithm 6 formally presents the process of Bayesian-based boosting for structured additive regression models.

In step 4 of algorithm 6, we derive the pseudo-residuals from the negative gradient of the L_2 loss, which inherits from the gradient boosting framework. From the loss perspective, in case of complex models like generalized additive models for location, scale and shape (GAMLSS), the loss function ρ is usually the negative log-likelihood of the assumed distribution of the response (Thomas et al., 2018; Zhang et al., 2022b). From the framework perspective, we can also treat the pseudo-residuals as the values derived from the likelihood-based boosting by taking the current additive predictor $\hat{\eta}^{[m-1]}$ as an offset. Regardless of how the pseudo-residuals are constructed (likelihood or negative gradient), the proposed Bayesian-based boosting does not deviate from the core concept of boosting, i.e. building a strong predictor by refitting the residuals.

In the step 9, we take the posterior mode as estimate. Typically, the mode is used for discrete variables, but it can still be estimated by a smooth function for continuous ones, which is exactly what we have done later in the simulation and application section. The choice of posterior statistics is subjective and depends on the concrete situation.

Algorithm 6 Bayesian-based boosting for structured additive regression

```
Initialize the additive predictor with a starting value, e.g. $\hat{\eta}^{[0]} = (\mathbf{0})_{i=1,\dots,n}$.
Initialize the hyperparameters $\tau^2, a, b$.
for Boosting iteration $m = 1, \dots, M$ do
    Calculate the negative gradient vector with for example the $L_2$ loss,
        $u^{[m]} = \left( -\frac{\partial}{\partial \eta} \rho(y, \eta) \Big|_{\eta = \hat{\eta}^{[m-1]}} \right)_{i=1,\dots,n} = y - \hat{\eta}^{[m-1]}.$
    for Each base-learner $p = 1, \dots, P$ do
        Construct the design matrix $\boldsymbol{Z}_p$.
        for MCMC samples $t = 1, \dots, T$ do
            Draw a sample of $\hat{\boldsymbol{\beta}}^{(t)}$ from $\mathrm{N}(\boldsymbol{\mu}_{\boldsymbol{\beta}_p}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_p})$ with
                $\boldsymbol{\Sigma}_{\boldsymbol{\beta}_p} = \left( \frac{1}{(\hat{\sigma}_p^2)^{(t)}} \boldsymbol{Z}_p^T \boldsymbol{Z}_p + \frac{1}{\tau^2} \boldsymbol{K} \right)^{-1},$
                $\boldsymbol{\mu}_{\boldsymbol{\beta}_p} = \frac{1}{(\hat{\sigma}_p^2)^{(t)}} \boldsymbol{\Sigma}_{\boldsymbol{\beta}_p} \boldsymbol{Z}_p^T u^{[m]},$
            where $\boldsymbol{K}$ is the corresponding penalty matrix.
            Draw a sample of $(\hat{\sigma}_p^2)^{(t)}$ from $\mathrm{IG}(\tilde{a}, \tilde{b})$ with
                $\tilde{a} = a + \frac{n}{2},$
                $\tilde{b} = b + \frac{1}{2} \left( u^{[m]} - \boldsymbol{Z}_p \hat{\boldsymbol{\beta}}_p^{(t)} \right)^T \left( u^{[m]} - \boldsymbol{Z}_p \hat{\boldsymbol{\beta}}_p^{(t)} \right).$
        end for
        Take the posterior modes as estimates,
            $\hat{\boldsymbol{\beta}}_p^{[m]} = \mathrm{mode}\{\hat{\boldsymbol{\beta}}_p^{(1)}, \dots, \hat{\boldsymbol{\beta}}_p^{(T)}\},$
            $\hat{\sigma}_p^{2[m]} = \mathrm{mode}\{(\hat{\sigma}_p^2)^{(1)}, \dots, (\hat{\sigma}_p^2)^{(T)}\}.$
        Calculate the model improvement
            $\mathrm{MSE}_p = \frac{1}{n} \sum_{i=1}^{n} \left( u_i^{[m]} - \boldsymbol{Z}_{ip} \hat{\boldsymbol{\beta}}_p^{[m]} \right)^2.$
    end for
    Select the best-performing $p^*$-th base-learner
        $p^* = \arg\min_p \mathrm{MSE}_p.$
    Update
        $\hat{\boldsymbol{\beta}}^{[m]} = \hat{\boldsymbol{\beta}}^{[m-1]} + \hat{\boldsymbol{\beta}}_{p^*}^{[m]}.$
end for
```

Other statistics such as mean or median are of course possible, but a different choice will not fundamentally change the results.

Similarly, the other parts of algorithm 6 are also established based on the commonly used specifications. The use of different preferred priors and model improvement criteria does not affect the effectiveness of the framework.

In usual gradient boosting algorithms, a step-length or learning-rate parameter is multiplied to the update $\hat{\boldsymbol{\beta}}_{p^*}^{[m]}$ in step 15. It is not necessary for the Bayesian-based boosting framework, since on one hand, the small steps are ensured by the variance hyperparameter τ^2 as discussed in section 4.1.2. On the other hand, additional shrinkage parameter will trigger a philosophical debate about whether uncertainty can be partitioned, but incorporating the shrinkage step into the parameter τ^2 avoids this discussion since the inference is always established on the complete and unpartitioned uncertainty.

4.1.5 Discussion of the uncertainty

For the last step in algorithm 6 (step 15), we arbitrarily focus on the p^*-th component instead of the entire $\boldsymbol{\beta}$ vector for convenience, i.e.

$$\hat{\beta}_{p^*}^{[m]} = \hat{\beta}_{p^*}^{[m-1]} + \hat{\beta}_{p^*,\text{update}}^{[m]}.$$

The expression of this equation indicates that $\hat{\beta}_{p^*}^{[m]}$ consists of two parts, the values at the previous iteration $\hat{\beta}_{p^*}^{[m-1]}$ and the update at the current iteration $\hat{\beta}_{p^*,\text{update}}^{[m]}$. Considering that $\hat{\beta}_{p^*}^{[m]}$ in Bayesian statistics is a random variable and that the value of the previous iteration has been determined, the conditional distribution of $\hat{\beta}_{p^*}^{[m]}$ is actually dominated by the distribution of the update $\hat{\beta}_{p^*}^{[m]}$, that is to say by

$$\begin{aligned} p(\hat{\beta}_{p^*}^{[m]} | \hat{\beta}_{p^*}^{[m-1]}) &= p(\hat{\beta}_{p^*}^{[m-1]} + \hat{\beta}_{p^*,\text{update}}^{[m]} | \hat{\beta}_{p^*}^{[m-1]}) \\ &= p(\hat{\beta}_{p^*,\text{update}}^{[m]} | \hat{\beta}_{p^*}^{[m-1]}). \end{aligned}$$

Given the Gaussian prior as above, the posterior distribution is then

$$\hat{\beta}_{p^*,\text{update}}^{[m]} | \hat{\beta}_{p^*}^{[m-1]} \sim \text{N}(\mu_{\beta_{p^*}}, \Sigma_{\beta_{p^*}}),$$

and this implies as well that the conditional distribution of $\hat{\beta}_{p^*}^{[m]}$ follow the same posterior distribution. Specifically, the conditional variance of $\hat{\beta}_{p^*}^{[m]}$ is given by

$$\mathrm{Var}(\hat{\beta}_{p^*}^{[m]}|\hat{\beta}_{p^*}^{[m-1]}) = \mathrm{Var}(\hat{\beta}_{p^*,\mathrm{update}}^{[m]}|\hat{\beta}_{p^*}^{[m-1]}) = \left(\frac{1}{\sigma^2}Z^T Z + \frac{1}{\tau^2}K\right)^{-1}. \tag{4.1}$$

If we compare this to the variance of the least square estimator in linear regression,

$$\mathrm{Var}(\hat{\beta}_{\mathrm{LS}}) = \left(\frac{1}{\sigma^2}Z^T Z\right)^{-1}, \tag{4.2}$$

we can easily find that their difference lies in the penalty component $\frac{1}{\tau^2}K$, and especially τ^2, since the penalty matrix K is usually fixed and has little flexibility. The conditional variance changes in the same direction as τ^2, that is, a larger τ^2 yields a larger $\mathrm{Var}(\hat{\beta}_{p^*}^{[m]}|\hat{\beta}_{p^*}^{[m-1]})$ and vice versa. In the extreme case of $\tau^2 \to \infty$, the conditional variance converges to the variance of the least square estimator.

Intuitively, it is reasonable that the conditional variance is affected by τ^2 in addition to the model variance σ^2, because even though we can treat τ^2 as a flexible penalty term, τ^2 is essentially a variance hyperparameter of the prior of β. If the variance in the prior is large, we cannot easily derive a small variance in the posterior, especially when only insufficient data or so-called evidence exists. Conversely, for a given evidence, a smaller τ^2 usually implies a stronger belief in the prior, allowing the posterior to vary only within a narrower interval, i.e. a smaller variance.

Similarly, by reviewing the entire $\boldsymbol{\beta}$ vector instead of that for the arbitrary p^*-th component, we can take the variance in the posterior in each component respective its last updated iteration as the measure of uncertainty. Thereby, uncertainty analysis in boosting is made possible.

4.2 Simulation

To test the effectiveness of the proposed algorithm `bboost` (short for Bayesian-based boosting) for structured additive regression models, we divide the simulation analysis into a linear and a non-linear regression part. For each case, the proposed approach will be properly compared with other commonly used methods, including classical linear regression models (`lm`) as well as non-linear approaches - the generalized additive models with integrated smoothness estimation (`gam`) (Wood, 2017), Bayesian inference (`BayesX`) (Umlauf et al., 2015; Belitz et al., 2022), and the gradient boosting with componentwise models, which is also referred as the model-based boosting algorithm (`mboost`) (Hothorn et al., 2010; Hofner et al., 2014; Hothorn et al., 2022).

4.2.1 Model setup

For the linear regression, we assume independent uniformly distributed covariates $X_j \sim \text{U}(-3,3), j \in \{1,\dots,p\}$ and a standard normal distributed error term $\epsilon_i, i \in \{1,\dots,300\}$. The realization of the response y depends only on the first four covariates,

$$y_i = \frac{1}{2} + 2x_{i1} + x_{i2} - 2x_{i3} - x_{i4} + \epsilon_i.$$

In addition to the four informative covariates, the input data contains also some non-informative ones: for $p = 4$, there exists no noise variable, for $p = 10$, there are low-dimensional 6 noise variables, and for $p = 500$, there are high-dimensional 496 noise variables, which is also the case of more covariates ($p = 500$) than observations ($n = 300$).

In terms of the model setup, `bboost` draws $T = 1000$ MCMC samples for the estimates in each iteration. Due to the fact that the proposed algorithm is not very computationally efficient, we use AIC instead of cross-validation to determine the stopping iteration and the maximal boosting iteration is limited to $M = 300$. The variance hyperparameter or shrinkage parameter τ^2 is element of the set of $\{0.00001, 0.001, 1, 1000\}$. Since we have a good experience with $\tau^2 = 0.001$, we take this value as default. In addition, both hyperparameters a and b in the inverse Gamma prior are set to the small value of 0.001. For other methods (`lm`, `BayesX` and `mboost`), we just take their default settings in `R` package to make inference. For `mboost`, the stopping iteration is determined by both 10-fold cross-validation and AIC to compare the results better. Finally, we perform 100 simulation runs to analyze the stabilized results.

For the case of non-linear regression, we similarly simulate six independent uniformly distributed covariates $X_p \sim \text{U}(-4,4), p \in \{1,\dots,6\}$ but only the first three are informative. Then, we simulate the response through

$$y_i = -\sin(x_{i1}) - \cos(x_{i2}) - \frac{1}{4}x_{i3}^2 + \epsilon_i, \tag{4.3}$$

with $\epsilon_i \sim \text{N}(0, 0.2), i \in \{1, \dots, 300\}$. Note that since the shape of the cosine function of x_2 in the input domain from -3 to 3 is similar to the square function, we draw samples from -4 to 4 to increase the complexity of the non-linear simulation. To test the estimation accuracy and the performance of variable selection, we set the input model as

$$y = s(x_1) + \cdots + s(x_6),$$

which specify a P-spline smooth term for each covariate. For `bboost`, the shrinkage parameter is $\tau^2 = 0.001$ and 1000 MCMC samples of the estimates are drawn in each iteration. Again, this simulation is rerun 100 times with different random seeds to stabilize outcomes for further analysis.

4.2.2 Linear regression

Since the construction of splines in non-parametric regression is relatively complicated, which makes the direct comparison of coefficients, especially the coefficients within splines, very hard, we feature the performance in regards to estimation accuracy and variable selection, as well as the general properties of `bboost`, in this section about linear regression.

The estimation accuracy is evaluated by the averaged (sum/mean) squared errors over the 100 simulation runs:

$$\text{SSE}_y := \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \text{and} \quad \text{MSE}_\beta := ||\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}||^2 \quad \text{and} \quad \text{SE}_{\sigma^2} := (\sigma^2 - \hat{\sigma}^2)^2,$$

where $||\cdot||$ denotes the Euclidean norm. The behavior of over- or underfitting is measured by the false positive rate (FP), which divides the number of false positives by the total number of noise variables. Since we practically observe no false negatives, they are omitted from the analysis.

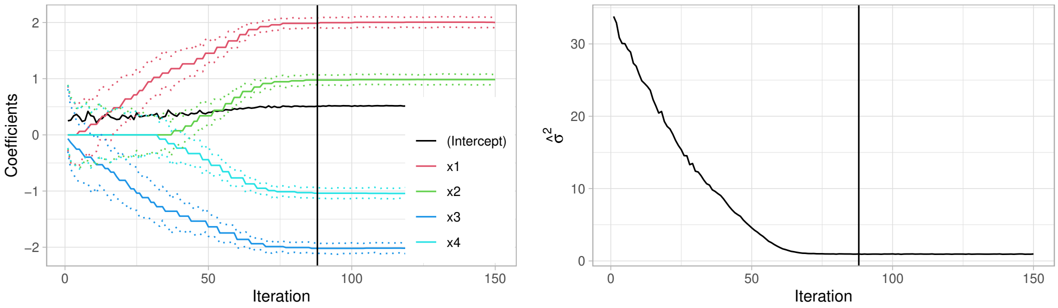
Table 4.1 lists the estimation accuracy of different approaches for various covari-

Table 4.1 Estimation accuracy of different methods averaged over 100 simulation runs. In Bayesian-based boosting (`bboost`), values are taken for the shrinkage parameter $\tau^2 = 0.001$. In linear models (`lm`), the false positive rate is calculated based on the 95% significance level.

p	Method	SSE$_y$	(SD)	MSE$_\beta$	(SD)	SE$_{\sigma^2}$	(SD)	FP	(SD)
4	`bboost`	293.799	19.282	0.173	0.045	0.003	0.006	-	-
	`lm`	293.292	19.343	0.170	0.044	0.004	0.007	-	-
	`BayesX`	295.951	23.644	0.171	0.043	0.007	0.007	-	-
	`mboost`(AIC)	293.296	19.344	0.170	0.044	0.005	0.008	-	-
	`mboost`(CV)	293.336	19.349	0.170	0.044	0.005	0.008	-	-
10	`bboost`	291.506	19.052	0.180	0.046	0.003	0.006	0.242	0.225
	`lm`	287.653	19.120	0.190	0.046	0.004	0.007	0.042	0.096
	`BayesX`	289.635	23.462	0.193	0.046	0.007	0.007	-	-
	`mboost`(AIC)	289.975	19.187	0.179	0.045	0.005	0.009	0.537	0.184
	`mboost`(CV)	289.388	19.445	0.181	0.045	0.005	0.009	0.628	0.273
500	`bboost`	225.767	21.679	0.239	0.056	0.057	0.032	0.055	0.016
	`mboost`(AIC)	44.074	13.261	0.779	0.103	0.729	0.073	0.458	0.031
	`mboost`(CV)	264.750	34.960	0.216	0.052	0.027	0.037	0.042	0.025

ates. It can be observed that the proposed approach `bboost` in general exhibits no noticeable difference from the other commonly used methods in all metrics. For the high-dimensional case ($p = 500$), in-sample sum squared errors (SSE$_y$) in `mboost`(AIC) are much smaller than `bboost` and `mboost`(CV), but it is obviously overfitted since not only its false positives are much bigger but also the squared errors for β and σ^2 underperform the other two. In the relatively more appropriate comparison between `bboost` and `mboost`(CV), it can be seen, that `SSE`$_y$ in the former are to some extend smaller than those in the latter at the cost of the inclusion of a few more false positive covariates. But for $p = 10$, at about the same level of squared error differences, `bboost` includes obviously less noise variables than `mboost`. Note that `lm` and `BayesX` contain only cases with no ($p = 4$) and low-dimensional ($p = 10$) noise variables, since they are not able to deal with data containing more variables than observations. In addition, due to the relatively bad variable selection experience with `mboost`(AIC) not only in the simulation study but also in the empirical analysis, we use the CV output as the default for `mboost` in the following context.

The results in table 4.1 indicate that the general performance of `bboost` is very competitive. The proposed method not only performs as well as the other commonly used methods, but it also maintains the core features of the other methods, i.e. it benefits from the uncertainty estimation of methods like `lm` and `BayesX` and it benefits



(a) Coefficient path **(b)** Model variance $\hat{\sigma}^2$ path

Figure 4.1 Convergence behavior of the `bboost` (with $\tau^2 = 0.001$) model. (a) shows the estimated coefficient path (solid curves) with 95% credible intervals (dashed curves). (b) shows the convergence path of the estimated model variance $\hat{\sigma}^2$. For both plots, the stopping iteration $m_{\text{stop}} = 88$ is marked with the vertical black line.

from the variable selection of boosting methods (`mboost`). In the following, we would like to further investigate the uncertainty of estimates and the effect of the shrinkage parameter on the uncertainty as stated in equation (4.1).

Figure 4.1 illustrates the convergence paths for the estimated coefficients from an arbitrary simulation chosen from the 100 reruns with $p = 10$, and its corresponding course of model variance. Firstly, as a member of the componentwise boosting family, the `bboost` model is able to construct the coefficient path just as the usual componentwise gradient boosting always does. Under the same scale, i.e. all variables are uniformly distributed from -3 to 3 in our case, the covariate with large coefficients tends to be included into the model at an early stage and followed by the ones with a smaller effect. As long as the model sufficiently extracts the information contained in the pseudo-residuals, the coefficients will converge to their corresponding stable regions, and the model variance will also converge. Secondly, as emphasized in the theoretical section, the main difference between `bboost` and existing boosting methods is the accessibility of estimation uncertainty. By reviewing the conditional variance in equation (4.1), it is clear that the uncertainty varies according to the model variance, and as $\hat{\sigma}^2$ converges, the uncertainty narrows to a smaller interval. In this example, beginning from an iteration near the stopping iteration (88-th), for example, the 75-th iteration, the true effects $(2, 1, -2, -1)$ for each covariate are already covered by the 95% credible intervals. Apparently, the availability of uncertainty allows us to double-check the quality of variable selection by viewing, for example, whether 0 lies outside of the interval.

To illustrate the effect of the shrinkage or the variance hyperparameter τ^2 on the

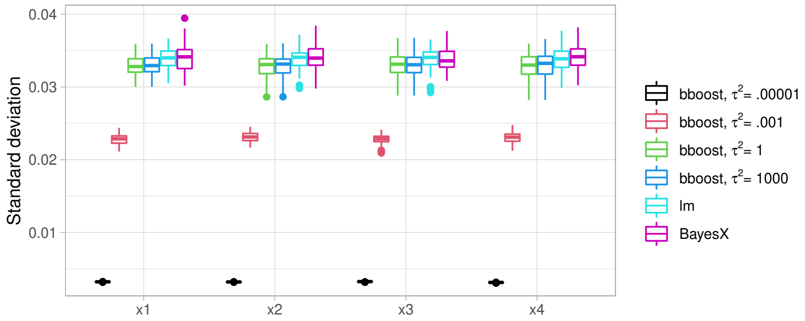


Figure 4.2 The summarized standard deviation of estimates for 100 simulation runs among different methods. For each simulation run, the standard deviation is calculated based on the MCMC samples for `BayesX` and `bboost` and based on the theoretical value for `lm`. For the case of $\tau^2 = 0.00001$, the values are taken at the pre-defined maximal stopping iteration.

uncertainty of the estimates, figure 4.2 summarizes the standard deviation of estimates in 100 simulation runs, and compares the `bboost` for $p = 10$ under different τ^2 with `lm` and `BayesX`. Since `mboost` provides no uncertainty information about the estimates, they are omitted from the figure.

As discussed in section 4.1.5, the uncertainty of `bboost` tends to show a smaller standard deviation than `lm` and `BayesX` due to the addition of the penalty component $\frac{1}{\tau^2}K$, and the larger τ^2 is, the closer the standard deviation of `bboost` is to the value of `lm`. Conversely, an extremely small τ^2 implies an almost unalterable firm belief in the prior, hence leading to a convergence of the uncertainty to zero. Note that the coefficients for the case of $\tau^2 = 0.00001$ do not converge at the given maximum stopping iteration due to the extremely small shrinkage. A larger τ^2 provides more similar information about uncertainty to the other methods, but it also reduces the impact of boosting, especially its shrinking and variable selection features, because the algorithm will stop very early if τ^2 is large.

4.2.3 Non-linear regression

Since the construction, re-parameterization and processing of smooth effects vary greatly among different methods for the case of non-linear regression, the direct comparison between coefficients is not informative, so here we pay more attention to variable selection and estimation accuracy, especially to prediction accuracy. But since the accuracy can be improved by adjusting parameters such as knots or degrees, the purpose

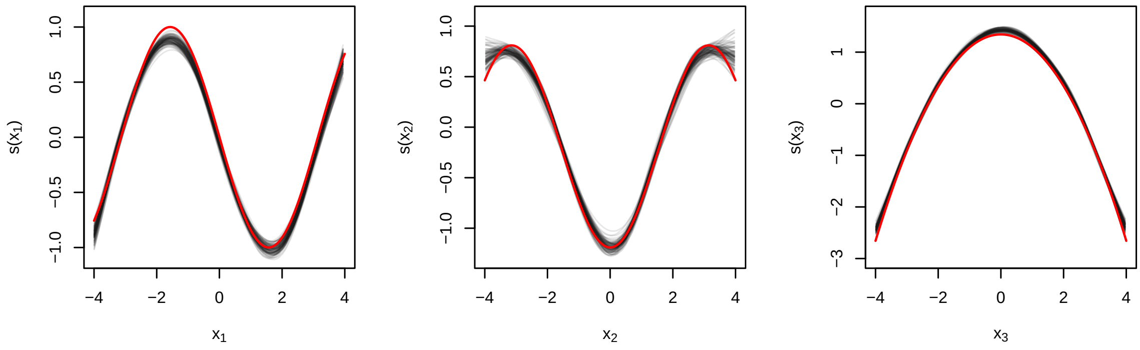


Figure 4.3 Smooth effects estimated by `bboost` for 100 simulation runs. The red curve in each subplot denotes the true effect and each light grey curve denotes the estimated effect in each simulation.

of accuracy comparisons is to highlight the relative difference rather than absolute values.

Firstly, figure 4.3 demonstrates estimation accuracy by showing the estimated smooth effects for the three informative covariate in 100 simulation runs. From this figure, we can clearly find that, even though `bboost` has some difficulties in estimating the points at, for example, the peaks for $s(x_1)$ and the edges for $s(x_2)$, it is able to discover the general effect of all three informative variables. Compared to the other algorithms, the biggest improvement of `bboost` lies in the effect selection. In our simulation setting, our proposed `bboost` has not only no false negatives, but also no false positives (therefore, the smooth effect for the three non-informative variables are excluded from the figure). In contrast, we observe a very high average false positive rate 0.977 for the conventional gradient boosting algorithm `mboost`. However, the cost of extremely low false positives is only a very slight increase in the in-sample MSE, which takes a value of 0.0385 for `bboost`, 0.0368 for `mboost`, and 0.0336 for `BayesX`. In addition, we observe also no false negatives from `mboost`.

The difficulties of `bboost` in making inferences for points at oscillation areas can also be observed for predictions on new data. Figure 4.4 selects one typical simulation run and compares the predictions between `bboost` and `BayesX` on newly generated 100 equidistant points from -4 to 4 for each covariate. Similar to the in-sample estimations as illustrated in figure 4.3, `bboost` has challenges in predicting points on edges, e.g. for x_2, since the range of credible intervals at these regions becomes larger, but the

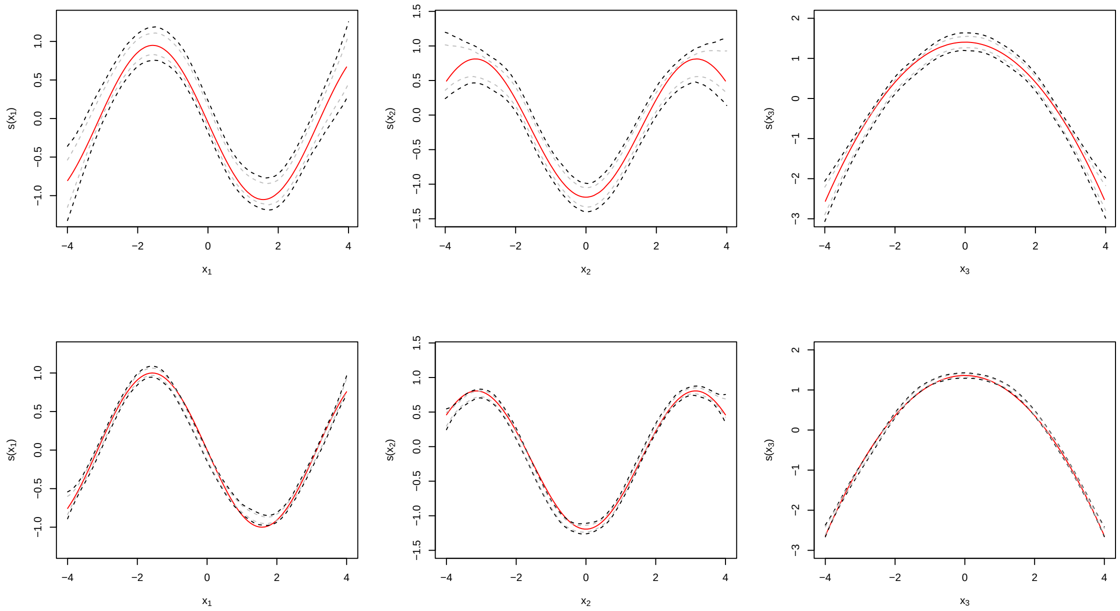


Figure 4.4 Comparison of credible intervals of the three informative covariate between `bboost` and `BayesX` for an arbitrary simulation run. The three plots at the top are the outcomes produced by `bboost` and the other three plots below are by `BayesX`. The red curve indicates the true effect, and the black and grey dashed lines are the 95% and 80% credible interval respectively.

intervals still cover the true values. In contrast, `BayesX` performs relatively better at edges and generates narrower prediction intervals throughout the whole input domain. However, narrower prediction intervals increase the risk of falsely excluding the truth, i.e. overfitting to some extent.

Figure 4.4 shows only the coverage of the truth for only one simulation. To get a better overview of the coverage behaviour, figure 4.5 summarizes the coverage rate at each point over 100 simulation runs, i.e. the number of runs in which the prediction interval covers the truth divided by the total 100 runs. In this figure, we can clearly find that the overall coverage rate for all the three covariates in `bboost` outperforms `BayesX` undoubtedly thank to the conservative prediction interval in `bboost`. In contrast, `BayesX` exhibits an imbalanced prediction quality, i.e. it is better at predicting values for example in the middle and edge of the input sequence than the ones at other areas. Narrower and more sensitive credible interval as illustrated in figure 4.4 indeed show overconfident predictions, which results in not only imbalanced prediction quality, but also in a worse overall coverage rate.

A potential reason for the relatively wider credible interval for `bboost` compared to

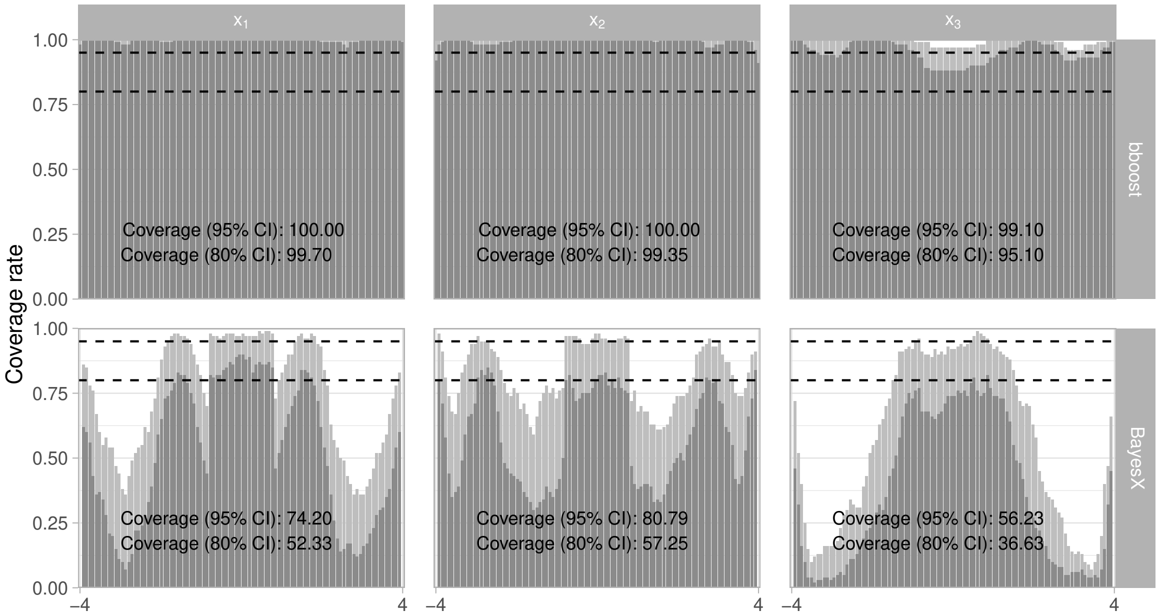


Figure 4.5 Coverage rates of predictions for the three covariates between `bboost` and `BayesX`. The dark and light grey area indicate the coverage rate of the 80% and 95% credible interval, respectively. The two horizontal dashed lines denote the 80% and 95% coverage rate. The overall coverage is labelled at the bottom.

`BayesX` is the exclusion of non-informative variables. In our setting, `BayesX` is not able to select variables, the noise ϵ in the response is partially explained by non-informative variables and hence results in narrower credible intervals for the informative effects. In contrast, due to the relatively good performance in variable selection, the effects for the informative variables have to take the noise ϵ into account and thus result in more uncertain estimates, i.e. wider credible intervals. However, wider interval in `bboost` does not imply worse estimation ability, but rather reflects a balance between estimation accuracy and variable selection.

Finally, we have to emphasize that the non-linear simulation here is mainly to show the differences between `bboost` and `BayesX`, but not the prediction accuracy. That is, both methods are able to capture the main pattern from the non-linear effects, thus they show almost no differences in, for example, mean squared errors, but `bboost` due to the variable selection tends to provide a more conservative prediction interval than `BayesX`.

4.3 Application

To further test the performance of the proposed Bayesian-based boosting for structured additive regression models, we take the Munich rent index data (Kneib et al., 2011) as a data example, which aims at predicting net rents based on a potentially large set of covariates, since this data not only contains a spatial effect, i.e. the location or district region affect rents, but also has many covariates that help to analyze the performance of variable selection.

The data were collected by Infratest Sozialforschung for the rental guide in the year of 2007 (for more details, see `http://www.muenchen.de/mietspiegel`). After data preprocessing such as the handling of missing values, validity check of covariates, etc., the final data contains 3019 flats and we include 200 covariates describing various characteristics of flats such as the living area, the quality of bathroom equipment, and the presence of central heating.

Previous analyses by Fahrmeir et al. (2021) have shown that the continuous variables living areas and the age of the building affect the net rent per square meter non-linearly. Moreover, a spatial variable indicating the district of flats in Munich is also available, which can be incorporated by using a Markov random field.

Therefore, we predict the net rents per square meter using the geoadditive regression model:

$$\text{rentsqm}_i = \beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta} + f_1(\text{area}_i) + f_2(\text{yearc}_i) + f_{\text{geo}}(\text{district}_i) + \epsilon_i,$$

where $\boldsymbol{x}_i$ denotes a 197-dimensional vector of mostly categorical covariates and $\boldsymbol{\beta}$ the corresponding effects. The non-linear functions f_1 and f_2 are cubic P-splines and f_{geo} constructs a Markov random field on the administrative districts in Munich. For `bboost`, the shrinkage parameter $\tau^2 = 0.001$ and 1000 MCMC samples are drawn in each boosting iteration and the AIC is used as the stopping criterion. As a baseline reference, the model is also estimated using the gradient boosting for additive models (`mboost`), the stopping iteration of which is also determined by AIC.

First, we focus on the linear effects selection, since all the three non-linear effects are chosen by both methods. There are 107 and 118 out of a total of 197 input linear effects are selected by `bboost` and `mboost` respectively, and there are 89 common

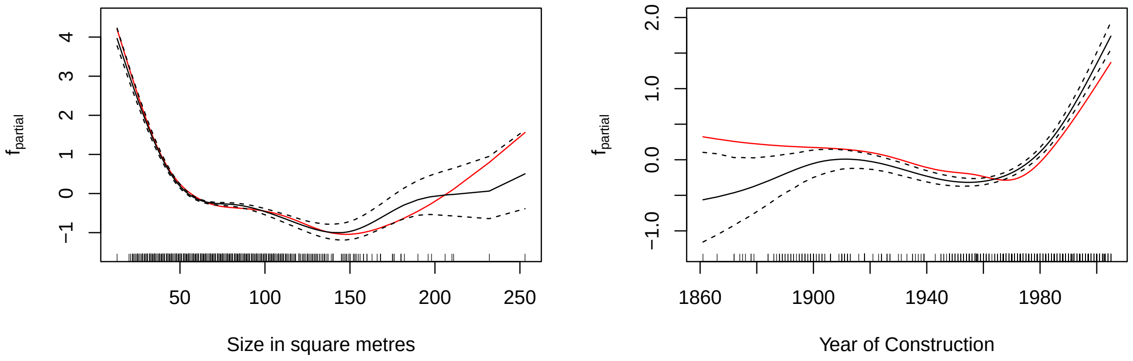
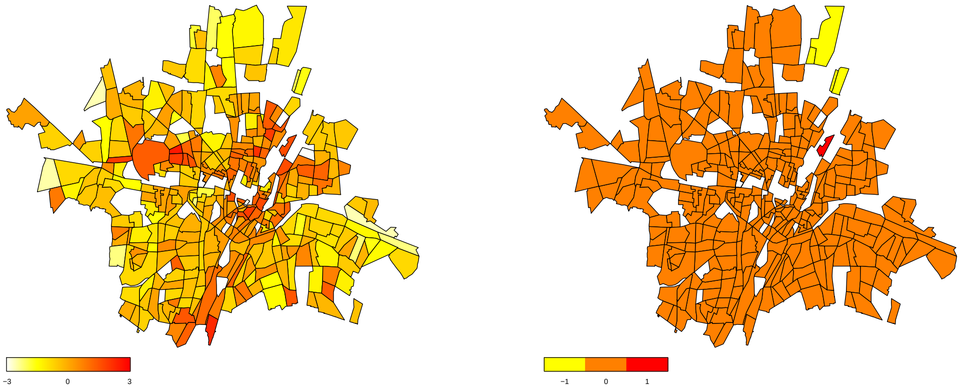


Figure 4.6 Comparison of smooth effects between `bboost` and `mboost`. The plot on the left shows the smooth effect of the variable living area and the right plot the variable year of construction. Estimates of `bboost` are depicted by black solid and dashed lines for the posterior mode and the 95% credible intervals, respectively, and estimates of `mboost` are depcited by red lines. The distribution of observations is illustrated through the rug at the bottom of each plot.

variables selected by both methods. In other words, about half the input linear effects are considered informative and `bboost` selects eleven fewer variables than `mboost`. Moreover, not only the majority of the selected covariates are common across the two, but also all the 89 commonly selected linear variables have the same coefficient sign, which shows that the proposed `bboost` can well capture the essential relationship between flat characteristics and rent. The basic views of `bboost` on the selection of effective variables and the direction of their influence are consistent with `mboost`.

Secondly, the comparison between both methods on the two smooth effects, living area and year of construction, is demonstrated in figure 4.6. It can be observed that `bboost` coincides with `mboost` in most regions, especially for values with more observations, while their divergence becomes larger when only a few observations are available, for example the estimates for very large or very old flats. A potential interesting finding might be the effect of very old flats on rents. According to the outcome of `mboost` and the study of Kneib et al. (2011), very old flats have little or some positive impact on rent due to, for example, their historical or relical value, whereas our approach demonstrates a negative impact, which might be due to the low renovation possibilities for these flats.

Finally, figure 4.7 illustrates the spatial effects estimated by both models. It can be expected that the flats at good location usually have higher rents, and this regular



(a) `bboost` **(b)** `mboost`

Figure 4.7 Spatial effects of the district variable on rents, where (a) shows the effects estimated by the `bboost` model and (b) the effects estimated by the `mboost` model.

pattern can be found by the outcomes from `bboost` model. The central areas, business districts, and areas adjacent to parks all have positive influence on rents, whereas the outlying districts show a clear negative impact. However, spatial effects are not shown as they should be in the `mboost` model. Except for a few districts, which demonstrate some negative and positive effects, the majority of regions show no differences. This indicates that although the region variable is selected into the final model, it still exhibits non-significant behavior. The variance in the rents can be sufficiently explained by the other linear and smooth effects.

Overall, the proposed `bboost` method has improved the variable selection to a certain extent under the premise of maintaining the fitting accuracy. Compared with the dogmatic method of gradient boosting in machine learning, the Bayesian-based boosting retains the advantages of uncertainty estimation in conventional statistical models, thereby leaving room for the error risk of prediction.

4.4 Summary and discussion

In this chapter, we extend the boosting framework based on Bayesian methodology by proposing a novel inference method, which implements the Bayesian inference in componentwise boosting. The proposed novel Bayesian-based boosting, on the one hand, retains the prior and uncertainty estimation of Bayesian inference. On the other hand, it also benefits from the shrinkage estimation and the intuitive variable selection procedure from boosting techniques.

The combination of Bayesian inference and boosting techniques enables analyses that were previously impossible with a single method, particularly analyzing the confidence intervals of estimated coefficients in the results of boosting and performing efficient and intuitive variable selection in Bayesian methods. The benefits of providing diverse information about the estimates are obvious, for example, one can achieve a more intuitive sense of the accuracy and bias of the estimates from boosting, and further uncertainty-based statistical analyses of estimates also become possible. Thereby, further statistical properties are added to the boosting algorithm attributed to the machine learning model, which is also an attempt to integrate machine learning and statistics at a deeper level.

The effectiveness of our proposed Bayesian-based boosting in these aspects are verified through simulation and empirical analyses. For the simulation of linear regression, the proposed novel algorithm realizes simultaneous uncertainty estimation and variable selection in the fitting process under the premise of ensuring accuracy. In the case of non-linear regression, our new method achieves a substantial improvement in variable selection at the cost of a very low loss of accuracy. However, it does not imply the preference for one of any aspect, but rather a balance between the estimation accuracy and effect selection.

The proposed algorithm provides a flexible Bayesian-based boosting framework for structured additive regression models. This means it leaves considerable space for further adjustment and improvement, for example, the commonly used priors in this chapter can surly be replaced with other priors depending on specific demands. Moreover, the Gibbs sampling used in the proposed algorithm can also be replaced with Metropolis-Hastings or other MCMC procedures.

Usually, a burn-in should be considered when using MCMC sampling. We omit this in the proposed Bayesian-based framework due to the intrinsic mechanism of boosting methods, that is, the best-fitting stopping iteration not only accounts for the complexity of the model but also for the quality of the fit, and sufficient fit usually implies the converge of coefficients. Nevertheless, samples drawn from the theoretical burn-in period have a very large variance and posterior modes thus oscillate heavily during these iterations, which does not meet the converge requirement. Conversely, the estimates at the stopping iteration are usually the coefficients that have been sufficiently estimated and the sufficient estimates also mean that the samples drawn in the nearby iteration have passed the burn-in period. In boosting methods, all estimates, including the samples in our proposed Bayesian-based boosting, before the stopping iteration are actually considered as burn-in. Therefore, an additional burn-in for the samples at the stopping iteration is completely unnecessary. However, due to the stochasticity of sampling, there still exits a very low probability that some samples in early iterations cause the AIC to be abnormally small, so it is judged that boosting is stopped early. It is thus recommended to force the stopping iteration to begin after some early iterations.

In terms of the uncertainty of estimates at the stopping iteration, they are actually the variation at their own last updates and have a close relationship with the variance of their least square estimator. This is easily reminiscent of the possibility for uncertainty analysis in conventional gradient boosting methods. In gradient boosting, the coefficients are usually estimated using least squares on the pseudo-residuals. However, the variance of the least square estimator, as suggested in equation (4.2), does not depend on the absolute values of pseudo-residuals, but on the model variance σ^2, which can also be regarded as the variance of the pseudo-residuals. Therefore, it seems possible to take the variance of the least square estimator for each variable at every last update as the uncertainty estimates in the usual gradient boosting. This would make the uncertainty analysis in the existing dogmatic boosting methods possible.

Another point that deserves further optimization is the complexity of the algorithm. There are three for loops nested in the proposed algorithm. Although it can be computed in parallel for each base-learner, looping MCMC samples inside boosting iterations is an immense computational burden, not to mention the requirement to tune the shrinking parameter and other refinements.

Chapter 5

Conclusion and Afterthoughts

In this thesis, the boosting framework is extended in several ways so that it becomes more powerful in fitting additive models. In the case of complex models, the existing boosting algorithms are enhanced, and for the general additive regression models, the boosting framework is extended based on Bayesian methodology.

5.1 Summary of the thesis

In Chapter 2, the imbalanced updates of predictors when applying boosting algorithm to complex models like GAMLSS are intensively discussed. The original cyclical componentwise gradient boosting algorithm for GAMLSS does not take the complexity across the prediction functions into account, so the non-cyclical approach is proposed in a way that the multiple-dimensional optimization problem of the stopping iteration reduces to a one-dimensional optimization procedure, which vastly reduces computing complexity. However, the achieved loss reduction of different distribution parameters cannot be addressed well by using a fixed step-length, so the adaptive step-length is suggested to ensure a fair selection. For the special case of Gaussian location and scale models, an an analytical solution for the adaptive step-length for the location parameter is derived, which avoids numerical optimization. As for the scale parameter, even though the exact analytical expression cannot be found, an approximate solution is suggested, which gives a better motivated default value than the commonly used value of 0.1.

The adaptive step-length discussed in Chapter 2 aims at enhancing the ability of boosting in complex models. The discussion on the other weakness of boosting in quantifying uncertainty of estimates begins from Chapter 3. In this section, a preliminary Bayesian-based boosting is proposed for linear mixed models, which divides the estimation procedure into two parts: the first estimates fixed effects still with componentwise gradient boosting, and the other estimates random effects with Bayesian inference. Thus, the variable selection feature of boosting is preserved and the uncertainty of random effects is accessible. The attempt of the fusion of boosting and Bayesian concepts discussed in this chapter focuses only on the specific linear mixed models and can only quantify uncertainty for the random effects part. The complete Bayesian-based boosting framework is presented in Chapter 4.

In Chapter 4, the proposed Bayesian-based Boosting is applied to the structured additive regression models, which contain not only the linear and random effects as the linear mixed models do, but it can also deal with smooth, spatial and other types of effects. Bayesian-based boosting is achieved by implementing Bayesian penalized regression in the componentwise boosting framework. The variance hyperparameter of the coefficient prior also plays the role of a shrinking parameter in boosting, but it avoids the problem of how to aggregate piecewise uncertainty in each iteration when directly using step-length as shrinkage, since the uncertainty of estimates is estimated separately in Bayesian-based boosting. The proposed Bayesian-based boosting is a powerful new approach, benefitting immensely from both the Bayesian and the boosting world.

5.2 Afterthoughts for further research

This thesis presents some new approaches and Bayesian-based boosting is possibly the first attempt in this area of boosting, so there are still some open problems and ideas for future research.

5.2.1 Further investigating analytical adaptive step-lengths

In Chapter 2, the adaptive step-length including its analytical as well as semi-analytical solutions in gradient boosting algorithm are intensively discussed for Gaussian location and shape models. Even though a numerical method for finding the optimal adaptive step-length is always a good choice, since the optimal value does not consistently have a closed form, it is still worth investigating the analytical solution for other distributions. On the one hand, compared to the numerical methods, the analytical solution will improve the computing efficiency, and on the other hand, it helps to reveal the inner relationships between the optimal step-length and the model parameters, which will give better suggestions for the commonly used but probably less than ideal step-length settings.

Another point that deserves further research is the correlation between the optimal step-length $\nu_{j^*,\boldsymbol{\mu}}^{*[m]}$ of a covariate and the coefficient of this covariate in the $\boldsymbol{\sigma}$-submodel. As illustrated in figure 2.7a, the converged adaptive step-lengths from high to low are the mother's BMI and age and the child's age and BMI, which matches the order the estimated coefficients of these variables in their $\boldsymbol{\sigma}$-submodel. Similar results are observed from the simulation study, although the results are not presented in this thesis. We tend to believe there exists a more precise mathematical dependence between the two terms.

5.2.2 Dependence between the prior coefficients and uncertainty

In the discussion of uncertainty of Bayesian-based boosting in Chapter 4, the dependence of the variance hyperparameter τ^2 and the variance of coefficients can be observed. The large τ^2 yields a large uncertainty of estimates. Theoretically, this dependence makes sense, since the posterior estimates have more uncertainty if there is less prior knowledge, and vice versa. Many applications prefer to use a non-informative prior, which usually comes with a larger variance hyperparameter compared to the default settings for $\tau^2 = 0.001$ used in this thesis. Nevertheless, the variance hyperparameter in the proposed Bayesian-based boosting framework has another role as the step-length or learning rate, while a large τ^2 will weaken the effectiveness of the boosting technique, and in extreme cases, the Bayesian-based boosting algorithm stops at the first step and the results coincide with the results of conventional Bayesian inference.

On the other side, if an extra shrinkage parameter is applied to the updates, the question arises whether the shrinkage should also be applied to the uncertainty of updates or not, since it raises a philosophical question of whether the uncertainty of an estimate is also divisible like the estimate itself in boosting. Our opinion is no, because if it could be possible, the aggregated uncertainty of estimates will approach infinity instead of a converged value. As illustrated in figure 4.1a, the variance of updates for arbitrary covariates converges to a constant, which is quite different from the fact that updates for coefficients converge to zero. The sum of a sequence approaching to zero converges to a constant, and the sum of a sequence approaching to a constant will go infinity. This means that the idea of enforcing the step-length by applying an additional shrinkage parameter and finally summing it up does not work for uncertainty, or at least it does not work in this way. Therefore, how to keep the effectiveness of Bayesian-based boosting for large τ^2 will be an interesting topic for further research.

5.2.3 Complexity of Bayesian-based boosting

The improvement of computing efficiency is always an important research topic. Instead of the negative attitude to look forward to the development of computers to achieve the purpose of improving efficiency, the optimization of the framework and the analytical solution or mathematical approximation to the steps are the practical way of thinking. As illustrated in algorithm 6, the proposed Bayesian-based boosting endures problem of complexity due to the three nested for loops, and the problem becomes sever in case of complex base-learners such as spatial effects. In addition, the complexity scales exponentially with the number of covariates and accounting for the tuning of model parameters, it is impractical to use Bayesian-based boosting to analyse large scale data.

Even though base-learners can be computed in parallel, it does not change the intrinsic nature of boosting and MCMC procedure, i.e. the pseudo-residuals can only be obtained after the finish of last boosting iteration, and likewise, the next MCMC sample is only available after the previous one has been drawn. The implementation of Bayesian inference in boosting framework makes the two method that are not so efficient more complicated. One possible solution is to replace the MCMC process with integrated nested Laplace approximations (INLA), but more works are needed to get a better knowledge about the effectiveness and efficiency as well as the bias induced by the approximation.

Bibliography

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory*, pages 267–281. Akadémiai Kiadó.

Allen, D. M. (1974). The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, 16(1):125–127.

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Bates, J., Pinheiro, J., Pinheiro, J., and Bates, D. (2000). *Mixed-Effects Models in S and S-PLUS*. Statistics and Computing. Springer New York.

Bauer, E. and Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1):105–139.

Belitz, C., Brezger, A., Kneib, T., Lang, S., and Umlauf, N. (2022). *BayesX: Software for Bayesian Inference in Structured Additive Regression Models*. Version 1.1.

Binder, H. (2013). CoxBoost: Cox models by likelihood based boosting for a single survival endpoint or competing risks. *R package version 1.4*, 1(4).

Binder, H. and Schumacher, M. (2008). Allowing for mandatory covariates in boosting estimation of sparse high-dimensional survival models. *BMC bioinformatics*, 9(1):1–10.

Brent, R. P. (2013). *Algorithms for minimization without derivatives*. Courier Corporation.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American statistical Association*, 88(421):9–25.

Breslow, N. E. and Lin, X. (1995). Bias correction in generalised linear mixed models with a single component of dispersion. *Biometrika*, 82(1):81–91.

Brezger, A. and Lang, S. (2006). Generalized structured additive regression based on Bayesian p-splines. *Computational Statistics & Data Analysis*, 50(4):967–991.

Brockhaus, S., Scheipl, F., Hothorn, T., and Greven, S. (2015). The functional linear array model. *Statistical Modelling*, 15(3):279–300.

Bühlmann, P. (2006). Boosting for high-dimensional linear models. *The Annals of Statistics*, 34(2):559–583.

Bühlmann, P. and Hothorn, T. (2007). Boosting algorithms: Regularization, prediction and model fitting. *Statistical science*, 22(4):477–505.

Bühlmann, P., Kalisch, M., and Meier, L. (2014). High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1(1):255–278.

Bühlmann, P. and Yu, B. (2003). Boosting with the L_2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339.

De Bin, R. (2016). Boosting in Cox regression: a comparison between the likelihood-based and the model-based approaches with focus on the R-packages CoxBoost and mboost. *Computational Statistics*, 31(2):513–531.

Dezeure, R., Bühlmann, P., Meier, L., and Meinshausen, N. (2015). High-dimensional inference: confidence intervals, p-values and r-software hdi. *Statistical science*, pages 533–558.

Elkan, C. et al. (1997). Boosting and naive Bayesian learning. Technical report, University of California, San Diego.

Fahrmeir, L. and Kneib, T. (2011). *Bayesian Smoothing and Regression for Longitudinal, Spatial and Event History Data*. Oxford University Press.

Fahrmeir, L., Kneib, T., and Lang, S. (2004). Penalized structured additive regression for space-time data: a Bayesian perspective. *Statistica Sinica*, pages 731–761.

Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. D. (2021). *Regression: Models, Methods and Applications*. Springer, Berlin, Heidelberg.

Fahrmeir, L. and Lang, S. (2001). Bayesian inference for generalized additive mixed models based on markov random field priors. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 50(2):201–220.

Fithian, W., Sun, D., and Taylor, J. (2014). Optimal inference after model selection. *arXiv preprint arXiv:1410.2597*.

Fong, Y., Rue, H., and Wakefield, J. (2010). Bayesian inference for generalized linear mixed models. *Biostatistics*, 11(3):397–412.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Greven, S. and Kneib, T. (2010). On the behaviour of marginal and conditional AIC in linear mixed models. *Biometrika*, 97(4):773–789.

Griesbach, C., Groll, A., and Bergherr, E. (2021a). Addressing cluster-constant covariates in mixed effects models via likelihood-based boosting techniques. *Plos one*, 16(7):e0254178.

Griesbach, C., Säfken, B., and Waldmann, E. (2021b). Gradient boosting for linear mixed models. *The International Journal of Biostatistics*, 17(2):317–329.

Groll, A. and Tutz, G. (2012). Regularization for generalized additive mixed models by likelihood-based boosting. *Methods of Information in Medicine*, 51(02):168–177.

Groll, A. and Tutz, G. (2014). Variable selection for generalized linear mixed models by L_1-penalized estimation. *Statistics and Computing*, 24(2):137–154.

Gumedze, F. and Dunne, T. (2011). Parameter estimation and inference in the linear mixed model. *Linear algebra and its applications*, 435(8):1920–1944.

Hastie, T., Taylor, J., Tibshirani, R., and Walther, G. (2007). Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1:1–29.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Heagerty, P. J. and Kurland, B. F. (2001). Misspecified maximum likelihood estimates and generalised linear mixed models. *Biometrika*, 88(4):973–985.

Hepp, T., Schmid, M., Gefeller, O., Waldmann, E., and Mayr, A. (2016). Approaches to regularized regression–a comparison between gradient boosting and the lasso. *Methods of information in medicine*, 55(05):422–430.

Hepp, T., Schmid, M., and Mayr, A. (2019). Significance tests for boosted location and scale models with linear base-learners. *The International Journal of Biostatistics*, 15(1).

Higham, N. J. (2002). Computing the nearest correlation matrix—a problem from finance. *IMA journal of Numerical Analysis*, 22(3):329–343.

Hofner, B., Boccuto, L., and Göker, M. (2015). Controlling false discoveries in high-dimensional situations: boosting with stability selection. *BMC bioinformatics*, 16(1):1–17.

Hofner, B., Mayr, A., Fenske, N., and Schmid, M. (2018). *gamboostLSS: Boosting Methods for GAMLSS Models*. R package version 2.0-1.

Hofner, B., Mayr, A., Robinzonov, N., and Schmid, M. (2014). Model-based boosting in R: a hands-on tutorial using the R package mboost. *Computational statistics*, 29(1):3–35.

Hofner, B., Mayr, A., and Schmid, M. (2016). gamboostLSS: An R package for model building and variable selection in the GAMLSS framework. *Journal of Statistical Software*, 74(1):1–31.

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., and Hofner, B. (2010). Model-based boosting 2.0. *The Journal of Machine Learning Research*, 11:2109–2113.

Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., and Hofner, B. (2022). *mboost: Model-Based Boosting.* R package version 2.9-7.

Hothorn, T., Müller, J., Schröder, B., Kneib, T., and Brandl, R. (2011). Decomposing environmental, spatial, and spatiotemporal components of species distributions. *Ecological Monographs*, 81(2):329–347.

Javanmard, A. and Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909.

Kneib, T., Konrath, S., and Fahrmeir, L. (2011). High dimensional structured additive regression models: Bayesian regularization, smoothing and predictive performance. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 60(1):51–70.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics*, 38(4):963–974.

Liang, H., Wu, H., and Zou, G. (2008). A note on conditional AIC for linear mixed-effects models. *Biometrika*, 95(3):773–778.

Lin, L., Drton, M., and Shojaie, A. (2020). Statistical significance in high-dimensional linear mixed models. In *Proceedings of the 2020 ACM-IMS on Foundations of Data*

Science Conference, FODS '20, page 171–181, New York, NY, USA. Association for Computing Machinery.

Lin, X. and Zhang, D. (1999). Inference in generalized additive mixed modelsby using smoothing splines. *Journal of the Royal Statistical Society: Series b (Statistical Methodology)*, 61(2):381–400.

Litière, S., Alonso, A., and Molenberghs, G. (2008). The impact of a misspecified random-effects distribution on the estimation and the performance of inferential procedures in generalized linear mixed models. *Statistics in medicine*, 27(16):3125–3144.

Lorbert, A., Blei, D. M., Schapire, R. E., and Ramadge, P. J. (2012). A Bayesian boosting model. *arXiv preprint arXiv:1209.1996*.

Mayr, A., Binder, H., Gefeller, O., and Schmid, M. (2014). The evolution of boosting algorithms. *Methods of information in medicine*, 53(06):419–427.

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(3):403–427.

Mayr, A., Hofner, B., Waldmann, E., Hepp, T., Meyer, S., and Gefeller, O. (2017a). An update on statistical boosting in biomedicine. *Computational and mathematical methods in medicine*, 2017.

Mayr, A., Schmid, M., Pfahlberg, A., Uter, W., and Gefeller, O. (2017b). A permutation test to analyse systematic bias and random measurement errors of medical devices via boosting location and scale models. *Statistical Methods in Medical Research*, 26(3):1443–1460.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473.

Meinshausen, N., Meier, L., and Bühlmann, P. (2009). P-values for high-dimensional regression. *Journal of the American Statistical Association*, 104(488):1671–1681.

Nock, R. and Sebban, M. (2001). A Bayesian boosting theorem. *Pattern Recognition Letters*, 22(3-4):413–419.

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(3):507–554.

Ripley, B. D. (2004). Selecting amongst large classes of models. In *Methods and Models in Statistics: In Honour of Professor John Nelder, FRS*, pages 155–170. World Scientific.

Rügamer, D., Brockhaus, S., Gentsch, K., Scherer, K., and Greven, S. (2018). Boosting factor-specific functional historical models for the detection of synchronization in bioelectrical signals. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(3):621–642.

Rügamer, D. and Greven, S. (2020). Inference for L_2-boosting. *Statistics and Computing*, 30(2):279–289.

Säfken, B., Rügamer, D., Kneib, T., and Greven, S. (2021). Conditional model selection in mixed-effects models with cAIC4. *Journal of Statistical Software*, 99(8):1–30.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, 5(2):197–227.

Schelldorfer, J., Bühlmann, P., and de Geer, S. v. (2011). Estimation for high-dimensional linear mixed-effects models using l_1-penalization. *Scandinavian Journal of Statistics*, 38(2):197–214.

Schmid, M. and Hothorn, T. (2008). Boosting additive models using component-wise P-splines. *Computational Statistics & Data Analysis*, 53(2):298–311.

Stasinopoulos, D. M. and Rigby, R. A. (2008). Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, 23:1–46.

Stasinopoulos, M. D., Rigby, R. A., Heller, G. Z., Voudouris, V., and De Bastiani, F. (2017). *Flexible regression and smoothing: using GAMLSS in R.* CRC Press.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133.

Stone, M. (1977). An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):44–47.

Thomas, J., Hepp, T., Mayr, A., and Bischl, B. (2017). Probing for sparse and fast variable selection with model-based boosting. *Computational and mathematical methods in medicine*, 2017.

Thomas, J., Mayr, A., Bischl, B., Schmid, M., Smith, A., and Hofner, B. (2018). Gradient boosting for distributional regression: faster tuning and improved variable selection via noncyclical updates. *Statistics and Computing*, 28(3):673–687.

Tibshirani, R. J., Taylor, J., Lockhart, R., and Tibshirani, R. (2016). Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, 111(514):600–620.

Ting, K. M. and Zheng, Z. (1999). Improving the performance of boosting for naive Bayesian classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 296–305. Springer.

Tutz, G. and Binder, H. (2006). Generalized additive modeling with implicit variable selection by likelihood-based boosting. *Biometrics*, 62(4):961–971.

Tutz, G. and Binder, H. (2007). Boosting ridge regression. *Computational Statistics & Data Analysis*, 51(12):6044–6059.

Tutz, G. and Groll, A. (2010). Generalized linear mixed models based on boosting. In *Statistical Modelling and Regression Structures*, pages 197–215. Springer.

Umlauf, N., Adler, D., Kneib, T., Lang, S., and Zeileis, A. (2015). Structured additive regression models: An R interface to BayesX. *Journal of Statistical Software*, 63(21):1–46.

Vaida, F. and Blanchard, S. (2005). Conditional Akaike information for mixed effects models. *Corrado Lagazio, Marco Marchi (Eds)*, page 101.

Wood, S. (2017). *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2 edition.

Yang, F., Foygel Barber, R., Jain, P., and Lafferty, J. (2016). Selective inference for group-sparse linear models. *Advances in neural information processing systems*, 29.

Zhang, B., Griesbach, C., and Bergherr, E. (2022a). Bayesian learners in gradient boosting for linear mixed models. *The International Journal of Biostatistics*.

Zhang, B., Hepp, T., Greven, S., and Bergherr, E. (2022b). Adaptive step-length selection in gradient boosting for Gaussian location and scale models. *Computational Statistics*, pages 1–38.

Zhang, T. and Yu, B. (2005). Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33(4):1538–1579.

Zhao, P. and Yu, B. (2004). Boosted lasso. Technical report, Department of Statistics, UC Berkeley.

Zhao, Y., Staudenmayer, J., Coull, B. A., and Wand, M. P. (2006). General design Bayesian generalized linear mixed models. *Statistical science*, pages 35–51.

Appendix A

Appendix

A.1 Derive the analytical ASL for the Gaussian distribution

Take the negative log-likelihood as the loss function, the loss for Gaussian distribution can be displayed as

$$\begin{aligned}\rho\left(\boldsymbol{y}, \{\eta_{\boldsymbol{\mu}}, \eta_{\boldsymbol{\sigma}}\}\right) = &- \log\left[\frac{1}{\left(\sqrt{2\pi}\right)^{n}} \cdot \det\left(\operatorname{diag}\left(\exp\left(-\eta_{\boldsymbol{\sigma}}(\boldsymbol{X})\right)\right)\right) \cdot \exp\left(-\frac{1}{2}(\boldsymbol{y} - \eta_{\boldsymbol{\mu}}\left(\boldsymbol{X}\right))^{T} \cdot \right.\right.\\ &\left.\left. \cdot \operatorname{diag}\left(\exp\left(-2\eta_{\boldsymbol{\sigma}}(\boldsymbol{X})\right)\right) \cdot \left(\boldsymbol{y} - \eta_{\boldsymbol{\mu}}(\boldsymbol{X})\right)\right)\right]\\ =&\frac{n}{2}\log(2\pi) + \mathbf{1}_{n}^{T}\eta_{\boldsymbol{\sigma}}(\boldsymbol{X}) + \frac{1}{2}\left(\boldsymbol{y} - \eta_{\boldsymbol{\mu}}(\boldsymbol{X})\right)^{T}\operatorname{diag}\left(\exp\left(-2\eta_{\boldsymbol{\sigma}}(\boldsymbol{X})\right)\right)\left(\boldsymbol{y} - \eta_{\boldsymbol{\mu}}(\boldsymbol{X})\right).\end{aligned}$$

The negative partial derivatives for both distribution parameters in iteration m are then

$$\begin{aligned}\boldsymbol{u}_{\boldsymbol{\mu}}^{[m]} =& - \frac{\partial\rho\left(\boldsymbol{y}, \{\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}, \hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}\}\right)}{\partial\hat{\eta}_{\boldsymbol{\mu}}} \qquad \text{(A.1)}\\ =& \operatorname{diag}\left(\exp\left(-2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{X})\right)\right)\left(\boldsymbol{y} - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{X})\right), \qquad \text{(A.2)}\\ \boldsymbol{u}_{\boldsymbol{\sigma}}^{[m]} =& - \frac{\partial\rho\left(\boldsymbol{y}, \{\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}, \hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}\}\right)}{\partial\hat{\eta}_{\boldsymbol{\sigma}}} \qquad \text{(A.3)}\\ =& - \mathbf{1}_{n} + \operatorname{diag}\left(\left(\boldsymbol{y} - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{X})\right)^{T}\right) \cdot \qquad \text{(A.4)}\\ & \cdot \operatorname{diag}\left(\exp\left(-2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{X})\right)\right)\left(\boldsymbol{y} - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{X})\right). \qquad \text{(A.5)}\end{aligned}$$

Both $\boldsymbol{u}_{\boldsymbol{\theta}}^{[m]}, \boldsymbol{\theta} \in \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ can be regressed on the simple linear base-learner $h_{j^*,\boldsymbol{\theta}}^{[m]}(\boldsymbol{x}_{\cdot j^*})$, where j^* denotes the best-fitting variable.

$$\boldsymbol{u}_{\boldsymbol{\mu}}^{[m]} = \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(\boldsymbol{x}_{\cdot j^*}) + \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\mu}}^{[m]} \tag{A.6}$$

$$\boldsymbol{u}_{\boldsymbol{\sigma}}^{[m]} = \hat{h}_{j^*,\boldsymbol{\sigma}}^{[m]}(\boldsymbol{x}_{\cdot j^*}) + \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\sigma}}^{[m]}, \tag{A.7}$$

where $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\mu}}^{[m]}$ and $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\sigma}}^{[m]}$ denote the residuals in simple linear regression models.

A.1.1 Optimal step-length for $\boldsymbol{\mu}$

The analytical optimal step-length for $\boldsymbol{\mu}$ in iteration m is obtained by minimizing the empirical risk,

$$\begin{aligned}
\nu_{j^*,\boldsymbol{\mu}}^{*[m]} &= \arg\min_{\nu} \sum_{i=1}^{n} \rho\left(y_i, \{\hat{\eta}_{\boldsymbol{\mu}}^{[m]}(\boldsymbol{x}_{i\cdot}), \hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\}\right) \\
&= \arg\min_{\nu} \sum_{i=1}^{n} \rho\left(y_i, \{\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) + \nu \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*}), \hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\}\right) \\
&= \arg\min_{\nu} \sum_{i=1}^{n} -\log\left[\frac{1}{\sqrt{2\pi}\exp(\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot}))} \exp\left(-\frac{\left(y_i - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) - \nu \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{2\exp(2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot}))}\right)\right] \\
&= \arg\min_{\nu} \sum_{i=1}^{n} \left[\frac{1}{2}\log(2\pi) + \log(\hat{\sigma}_i^{[m-1]}) + \frac{\left(y_i - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) - \nu \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{2\sigma_i^{2[m-1]}}\right] \\
&= \arg\min_{\nu} \sum_{i=1}^{n} \frac{\left(y_i - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) - \nu \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{2\hat{\sigma}_i^{2[m-1]}},
\end{aligned}$$

Note that the expression $\hat{\sigma}_i^{2[m-1]}$ represents the square of the standard deviation in the previous boosting iteration, i.e. $\hat{\sigma}_i^{2[m-1]} = (\hat{\sigma}_i^{[m-1]})^2$. And according to the model specification $\hat{\sigma}_i^{[m-1]} = \exp(\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot}))$.

It can be shown, that the expression is a convex function, so the optimal value $\nu_{\boldsymbol{\mu}}^{*[m]}$ is accessed by letting the first order derivative equal zero,

$$\frac{\partial}{\partial \nu} \sum_{i=1}^{n} \frac{\left(y_i - \hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) - \nu \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{2\hat{\sigma}_i^{2[m-1]}}$$

$$
\begin{aligned}
&\overset{Eq.(\mathrm{A}.2)}{=} \frac{\partial}{\partial \nu} \sum_{i=1}^{n} \frac{\left(u_{\boldsymbol{\mu},i}^{[m]} \hat{\sigma}_i^{2[m-1]} - \nu \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{2\hat{\sigma}_i^{2[m-1]}} \\
&= \frac{\partial}{\partial \nu} \sum_{i=1}^{n} \left(\frac{1}{2} u_{\boldsymbol{\mu},i}^{2[m]} \hat{\sigma}_i^{2[m-1]} - \nu \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*}) u_{\boldsymbol{\mu},i}^{[m]} + \frac{\nu^2 \left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{2\hat{\sigma}_i^{2[m-1]}} \right) \\
&= \sum_{i=1}^{n} \left(-\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*}) + \nu \frac{\left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\hat{\sigma}_i^{2[m-1]}} \right) \overset{!}{=} 0 \\
&\Leftrightarrow \nu = \frac{\sum_{i=1}^{n} \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*}) u_{\boldsymbol{\mu},i}^{[m]}}{\sum_{i=1}^{n} \frac{\left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\hat{\sigma}_i^{2[m-1]}}} \\
&\qquad \overset{Eq.(\mathrm{A}.6)}{=} \frac{\sum_{i=1}^{n} \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*}) \left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*}) + \hat{\epsilon}_{\boldsymbol{\mu},i}^{[m]}\right)}{\sum_{i=1}^{n} \frac{\left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\hat{\sigma}_i^{2[m-1]}}} \\
&\qquad = \frac{\sum_{i=1}^{n} \left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2 + \sum_{i=1}^{n} \hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*}) \hat{\epsilon}_{\boldsymbol{\mu},i}}{\sum_{i=1}^{n} \frac{\left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\hat{\sigma}_i^{2[m-1]}}} \\
&\qquad = \frac{\sum_{i=1}^{n} \left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\sum_{i=1}^{n} \frac{\left(\hat{h}_{j^*,\boldsymbol{\mu}}^{[m]}(x_{ij^*})\right)^2}{\hat{\sigma}_i^{2[m-1]}}},
\end{aligned}
$$

where $\sum_{i=1}^{n} \hat{h}_{j^*\boldsymbol{\mu}}^{[m]}(x_{ij^*}) \hat{\epsilon}_{\boldsymbol{\mu},i} = 0$, because the residuals are uncorrelated with the fitted values.

A.1.2 Optimal step-length for $\boldsymbol{\sigma}$

The analytical optimal step-length for $\boldsymbol{\sigma}$ in iteration m is obtained by minimizing the empirical risk,

$$
\begin{aligned}
\nu_{\boldsymbol{\sigma}}^{*[m]} &= \underset{\nu}{\arg\min} \sum_{i=1}^{n} \rho\left(y_i, \{\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}), \hat{\eta}_{\boldsymbol{\sigma}}^{[m]}(\boldsymbol{x}_{i\cdot})\}\right) \\
&= \underset{\nu}{\arg\min} \sum_{i=1}^{n} \rho\left(y_i, \{\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot}), \hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) + \nu \hat{h}_{j^*,\boldsymbol{\sigma}}^{[m]}(x_{ij^*})\}\right) \\
&= \underset{\nu}{\arg\min} \sum_{i=1}^{n} -\log\left[\frac{1}{\sqrt{2\pi}\exp\left(\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot}) + \nu \hat{h}_{j^*,\boldsymbol{\sigma}}^{[m]}(x_{ij^*})\right)}\cdot\right.
\end{aligned}
$$

$$
\begin{aligned}
&\cdot \exp\left(-\frac{\left(y_i-\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\right)^2}{2\exp\left(2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})+2\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)}\right)\Bigg)\Bigg]\\
=&\arg\min_{\nu}\sum_{i=1}^{n}\frac{1}{2}\log(2\pi)+\sum_{i=1}^{n}\left(\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})+\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)+\\
&+\sum_{i=1}^{n}\frac{\left(y_i-\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\right)^2}{2\exp\left(2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})+2\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)}\\
=&\arg\min_{\nu}\sum_{i=1}^{n}\left(\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})+\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)+\sum_{i=1}^{n}\frac{\left(y_i-\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\right)^2}{2\exp\left(2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})+2\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)}.
\end{aligned}
$$

It can be shown, that the second order derivative of the expression is positive and thus the expression a convex function. Letting the first order derivative equal zero, we get

$$
\begin{aligned}
&\frac{\partial}{\partial\nu}\left[\sum_{i=1}^{n}\left(\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})+\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)+\sum_{i=1}^{n}\frac{\left(y_i-\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\right)^2}{2\exp\left(2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})+2\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)}\right]\\
=&\sum_{i=1}^{n}\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})-\sum_{i=1}^{n}\left(y_i-\hat{\eta}_{\boldsymbol{\mu}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\right)^2\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\exp\left(-2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})-2\nu\hat{h}^{[m]}_{\boldsymbol{\sigma}}(x_{ij^*})\right)\\
\overset{Eq.(A.5)}{=}&\sum_{i=1}^{n}\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})-\sum_{i=1}^{n}\frac{u^{[m]}_{\boldsymbol{\sigma},i}+1}{\exp\left(-2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})\right)}\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\exp\left(-2\hat{\eta}_{\boldsymbol{\sigma}}^{[m-1]}(\boldsymbol{x}_{i\cdot})-2\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)\\
=&\sum_{i=1}^{n}\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})-\sum_{i=1}^{n}\left(u^{[m]}_{\boldsymbol{\sigma},i}+1\right)\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\exp\left(-2\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)\\
\overset{Eq.(A.7)}{=}&\sum_{i=1}^{n}\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})-\sum_{i=1}^{n}\frac{\left(\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})+\hat{\epsilon}^{[m]}_{\boldsymbol{\sigma},i}+1\right)\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})}{\exp\left(2\nu\hat{h}^{[m]}_{j^*,\boldsymbol{\sigma}}(x_{ij^*})\right)}\overset{!}{=}0
\end{aligned}
$$

A.2 Additional simulation graphics

In this appendix, we present the results for some of the simulated examples in Sect. 2.3.1. Boxplot of the estimated coefficients are showed in Figure A.1 and Figure A.2. Figure A.3 illustrates the negative log-likelihood. The summary of stopping iterations m_{stop} is demonstrated in Figure A.4.

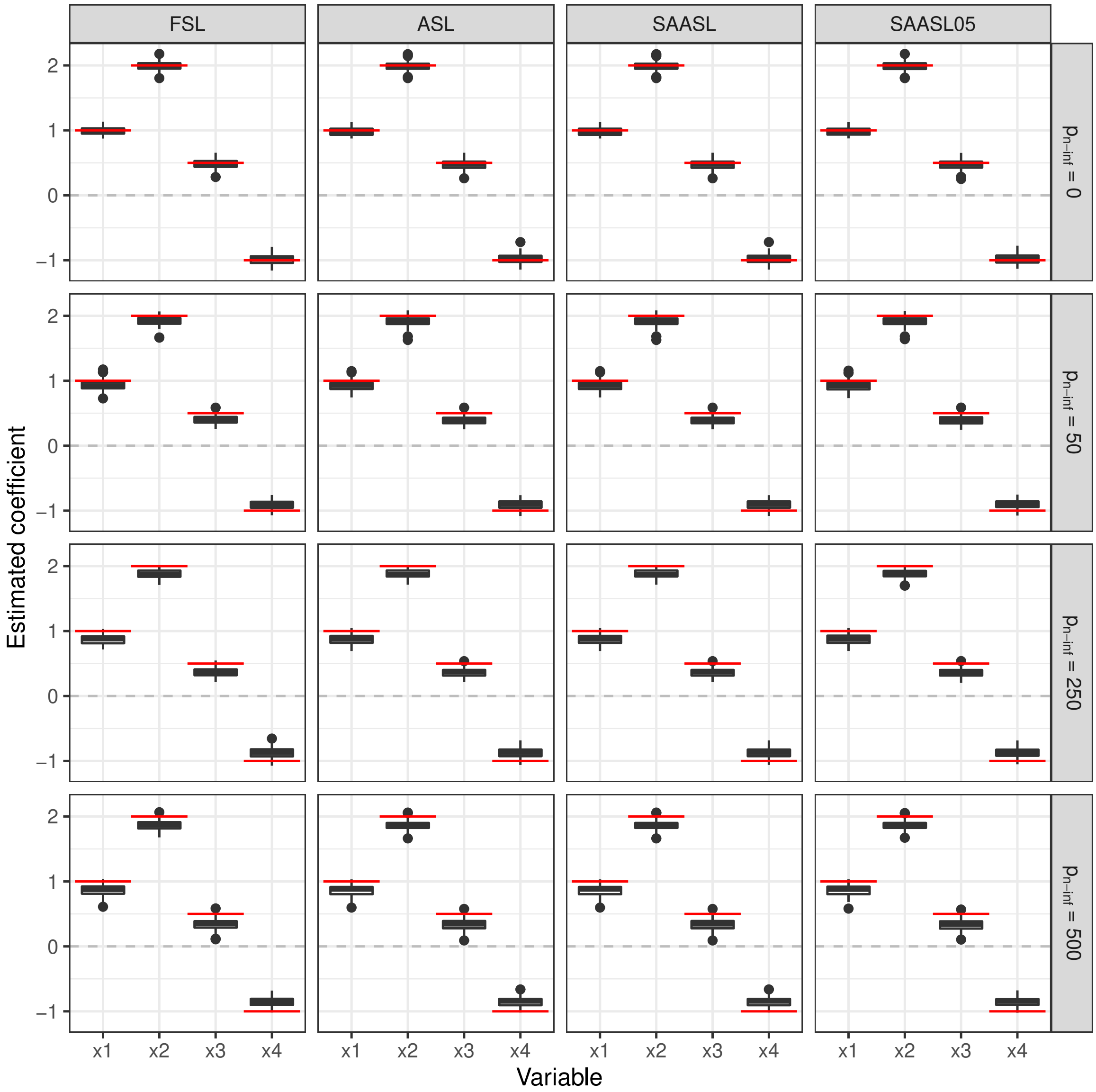


Figure A.1 Boxplot of the estimated coefficients of η_μ in 100 simulation runs. Values are taken at the stopping iterations determined by 10-folds cross-validation. The results are separated according to fixed and adaptive approaches with respect to different non-informative variables settings, i.e. $p_{\text{n-inf}} = 0, 50, 250$ and 500. The horizontal red lines indicate the true coefficients. The shrinkage of the coefficients towards zero can be observed from this graphic.

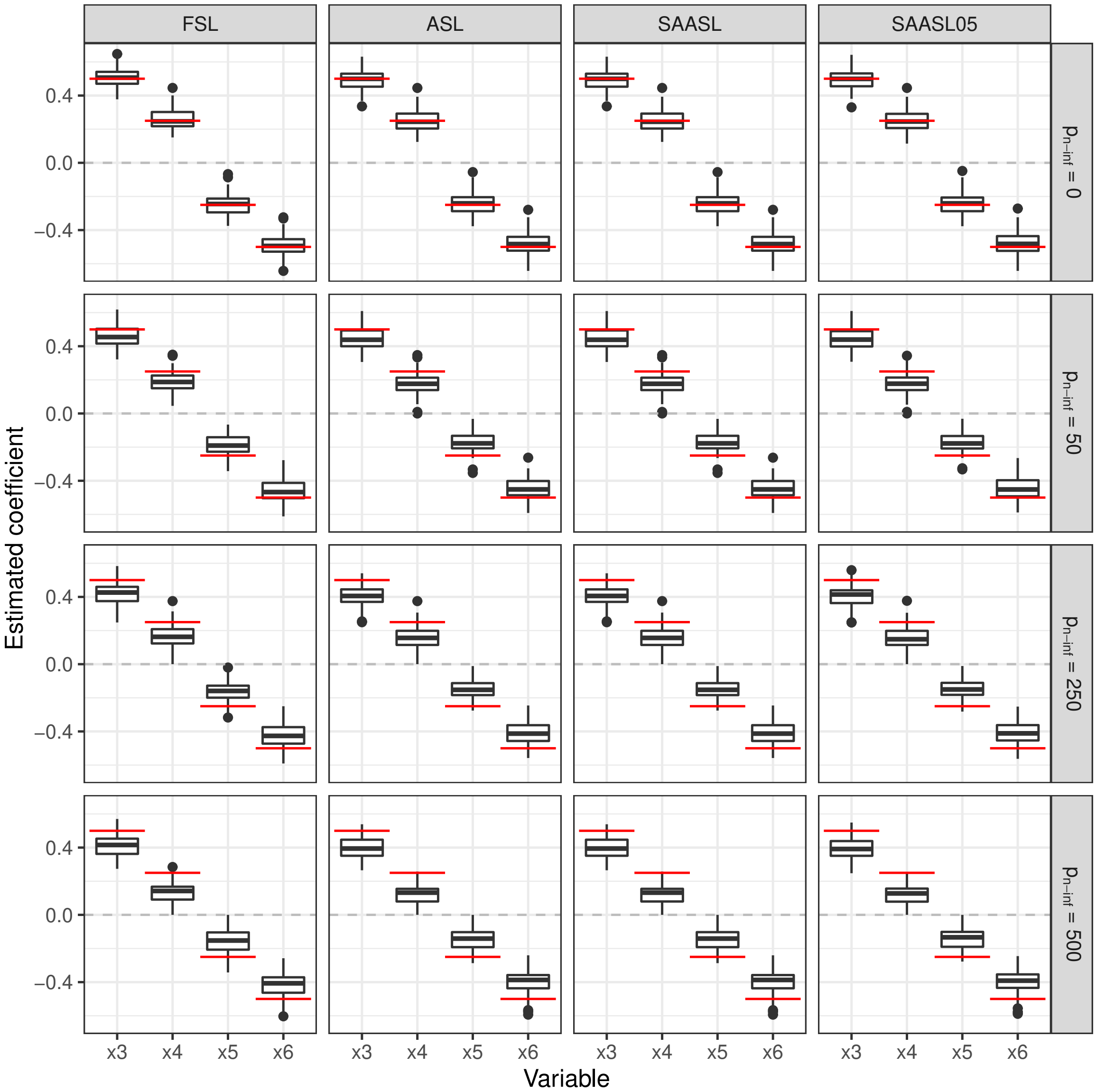


Figure A.2 Boxplot of the estimated coefficients of η_σ in 100 simulation runs. Values are taken at the stopping iterations tuned by 10-folds cross-validation. The results are separated according to fixed and adaptive approaches with respect to different non-informative variables settings, i.e. $p_{\text{n-inf}} = 0, 50, 250$ and 500. The horizontal red lines indicate the true coefficients. The shrinkage of the coefficients towards zero can be observed from this graphic.

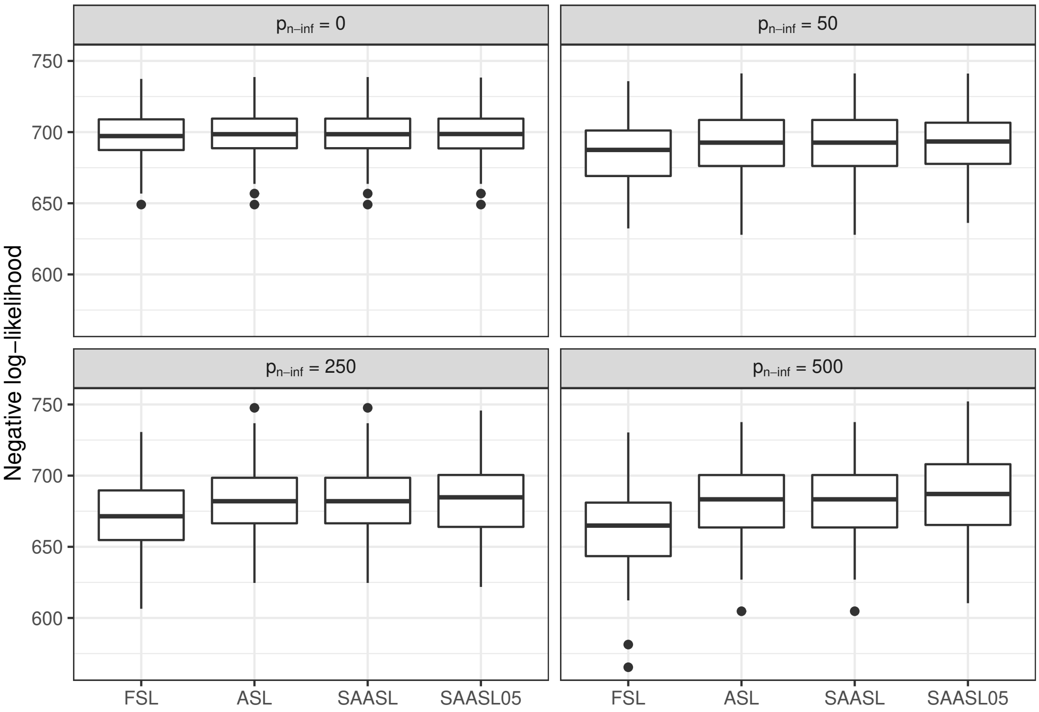


Figure A.3 Summary of the negative log-likelihood of 100 simulation runs with different estimating approaches with respect to various non-informative variables settings. Values are taken at the stopping iteration determined by 10-folds cross-validation.

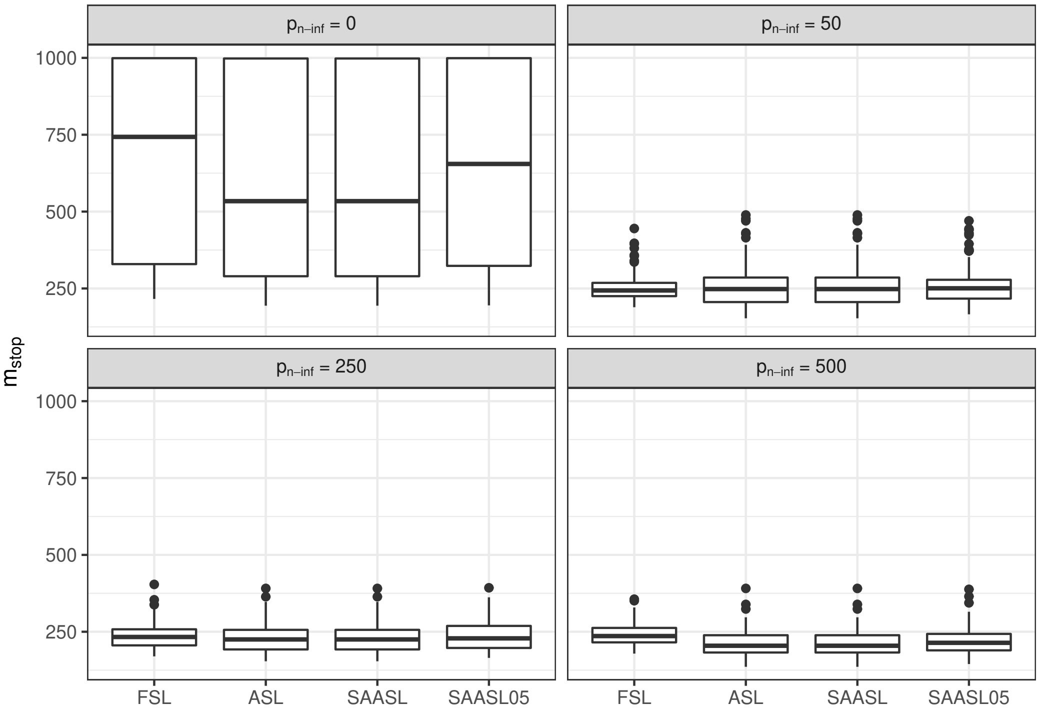


Figure A.4 m_{stop} tuned by 10-fold CV with different estimating methods with respect to different non-informative variables settings. The predefined maximal learning iteration is 1000.

A.3 Additional simulation table

The additional Table A.1 summaries the average MSE of the estimated coefficients for both Gaussian distribution parameters in Sect. 2.3.2.

Table A.1 The average MSE of the estimated coefficients for both model parameters $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ w.r.t. three estimation approaches. The MSE for each coefficient is calculated not only from 100 simulation runs (Total) at their stopping iterations but also from the true positive subsets (TP), i.e. the simulations from which a coefficient is selected by all three approaches.

	$\boldsymbol{\mu}$						$\boldsymbol{\sigma}$					
	$\hat{\beta}_1$		$\hat{\beta}_2$		$\hat{\beta}_3$		$\hat{\beta}_1$		$\hat{\beta}_2$		$\hat{\beta}_3$	
	Total	TP	Total	TP	Total	TP	Total	TP	Total	TP	Total	TP
FSL	13.7	80.9	26.5	112.2	13.8	76.0	0.84	0.81	1.41	1.42	0.84	0.82
GAMLSS	113.8	328.8	145.6	355.8	116.8	339.4	0.82	0.79	1.44	1.44	0.81	0.79
SAASL	71.3	250.2	95.8	271.7	73.3	238.5	0.85	0.81	1.39	1.40	0.85	0.82

A.4 Estimated coefficients of riboflavin dataset

In this appendix, we provide the estimated coefficients with fixed and adaptive approaches for riboflavin data in Sect. 2.4.2. Table A.2 and Table A.3 concern about the $\boldsymbol{\mu}$-submodel and $\boldsymbol{\sigma}$-submodel, respectively.

Table A.2 The estimated coefficients of the $\boldsymbol{\mu}$-submodel with fixed and adaptive approaches. Values are taken at the m_{stop} tuned by 5-folds CV.

	Variable	FSL	ASL	SAASL	SAASL05	glmnet
1	(Intercept)	-7.03	-7.04	-7.04	-7.03	0.72
2	ARGF_at	-0.08	-0.02	-0.02	-0.02	
3	IOLE_at	-0.32			-0.02	
4	LYSC_at					-0.06
5	RPLO_at				-0.02	-0.08
6	SPOIISA_at	0.35	0.19	0.19	0.23	0.02
7	XKDC_at	0.18			0.09	0.19
8	XKDO_at		0.02	0.02	0.03	
9	XKDS_at	0.11	0.08	0.08	0.08	0.06
10	XLYA_at					0.05
11	XTMA_at		0.03	0.03	0.01	
12	XTRA_at					0.01
13	YCDH_at					-0.03
14	YCGM_at	-0.09	-0.06	-0.06	-0.06	-0.01
15	YCGN_at		-0.04	-0.04	-0.04	
16	YCGO_at	-0.07				-0.14
17	YCGP_at				-0.03	
18	YCKE_at	0.15	0.12	0.12	0.14	0.15
19	YCLB_at	0.29				
20	YCSG_at		-0.08	-0.08	-0.18	
21	YDAO_at				-0.03	
22	YDAR_at	-0.24	-0.16	-0.16	-0.16	
23	YDDK_at					-0.04
24	YEBC_at					-0.55
25	YHAI_at		0.12	0.12	0.11	
26	YHFU_at		-0.02	-0.02	-0.03	-0.01
27	YJCJ_at	0.11	0.04	0.04	0.04	
28	YKBA_at					0.01
29	YKUH_at		0.05	0.05	0.06	
30	YOAB_at					-0.34
31	YORB_i_at		0.03	0.03	0.05	0.10
32	YOZH_i_at		0.02	0.02		
33	YPGA_at					-0.05
34	YTGB_at					-0.09
35	YWQD_at				-0.02	
36	YXJA_at		-0.01	-0.01	-0.01	
37	YXLC_at		-0.03	-0.03		
38	YXLD_at	-0.12	-0.14	-0.14	-0.16	-0.14
39	YXLE_at	-0.06	-0.01	-0.01	-0.01	

Table A.3 The estimated coefficients of $\boldsymbol{\sigma}$-submodel with fixed and adaptive approaches. Values are taken at the m_{stop} tuned by 5-folds CV.

	Variables	FSL	ASL	SAASL	SAASL05
1	(Intercept)	-1.41	-1.29	-1.29	-1.53
2	COTJC_at		-0.18	-0.18	
3	DEGA_at	-0.13	-0.61	-0.61	-0.89
4	EXPZ_at		0.21	0.21	0.05
5	LEVD_at	0.24	0.15	0.15	0.19
6	NTH_at	-0.09	-0.11	-0.11	
7	PHRI_r_at	0.06			0.03
8	TRUA_at	-0.09	-0.74	-0.74	-0.71
9	XLYA_at	-0.06			
10	XPF_at				-0.05
11	YACN_at	0.20			
12	YCNK_at	0.66	0.06	0.06	0.27
13	YFIG_at	-0.08			-0.09
14	YFMD_at	-0.25	-0.43	-0.43	-0.35
15	YHBD_at		-0.21	-0.21	-0.19
16	YHEN_at				-0.05
17	YHFS_at				0.06
18	YITQ_at				-0.24
19	YJFB_at		-0.11	-0.11	-0.09
20	YKRS_at		0.29	0.29	0.35
21	YKVV_at		0.06	0.06	0.28
22	YPGA_at	0.07	0.05	0.05	0.12
23	YSBA_at		-0.55	-0.55	-0.28
24	YSBB_at	-0.15	-0.23	-0.23	-0.20
25	YTFP_at	-0.24			
26	YTQI_at	0.10	0.16	0.16	0.11
27	YURR_at		-0.07	-0.07	-0.11
28	YWQA_at	-0.11			-0.20
29	YYAE_at				-0.06
30	YYBT_at	-0.08			-0.03

Acknowledgement

The thesis is the result of my research as a doctoral student at the Institute for Medical Informatics, Biometry and Epidemiology at the Friedrich-Alexander-Universität Erlangen-Nürnberg and at the Chair of Spatial Data Science and Statistical Learning at the Georg-August-Universität Göttingen. First of all, I would like to express my sincere gratitude to my supervisor Prof. Dr. Elisabeth Bergherr for her constant encouragement, motivation, and patient guidance. Her cheerful personality and optimistic attitude not only inspired my research work, but it also had a positive impact on my attitude towards life's challenges and my future career. Then, I would like to thank another supervisor, Dr. Tobias Hepp, for his willingness to spend a lot of time sharing his own research experiences and discussing with me on my right or false opinions.

I would also like to express my heartfelt thanks to Prof. Dr. Thomas Kneib, Prof. Dr. Sonja Greven, Prof. Dr. Andreas Mayr, and Dr. Colin Griesbach for their support and help with their erudite professional knowledge and experience in the formulation of this thesis. Furthermore, I want to thank my colleagues in Göttingen for the exchange and discussions on technical knowledge and the German word-of-the-day lunch lessons, as well as my colleagues in Erlangen for the cultural talks and sports activities.

Finally, I must express very profound gratitude to my family and my friends for providing me with unfailing support and continuous encouragement throughout these years.

Göttingen, December 2022 Boyao Zhang

Declaration

I herewith give assurance that I completed this dissertation independently without prohibited assistance of third parties or aids other than those identified in this dissertation. All passages that are drawn from published or unpublished writings, either word-for-word or in paraphrase, have been clearly identified as such. Third parties were not involved in the drafting of the content of this dissertation; most specifically, I did not employ the assistance of a dissertation advisor. No part of this thesis has been used in another doctoral or tenure process.

Place, Date

Signature

Curriculum Vitae

Personal Information

Name:	Boyao Zhang
Date of birth:	12.08.1989
Place of birth:	Heilongjiang, China
Marital status:	single
Nationality:	China

Education

since 10/2021	PhD student at the Georg-August-Universität Göttingen
09/2019 - 09/2021	PhD student at the Friedrich-Alexander-Universität Erlangen-Nürnberg
10/2016 - 06/2019	Student of statistics (M.Sc.) at the Ludwig-Maximilians-Universität München
10/2013 - 09/2016	Student of statistics (B.Sc.) at the Ludwig-Maximilians-Universität München
09/2008 - 06/2012	Student of statistics (B.Eco.) at the Hunan University
06/2008	Gaokao