

Introduction

- Variable selection and model choice are of major concern in statistical regression modeling. This is especially true if the number of covariates exceeds the number of observations in a data set (“ $p > n$ ”).
 - Which covariates should enter the model?
 - How should these covariates be modeled? (linear effects, smooth effects,...)
- Boosting is an established method for model fitting with intrinsic variable selection and model choice.
- A central problem remains: Variable selection and model choice are biased if covariates or modeling alternatives are of different nature.
- Idea: Prevent selection bias by making covariates and modeling alternatives comparable in the boosting framework.

Component-Wise Boosting

- Consider i.i.d. observations (y_i, \mathbf{x}_i) , $i = 1, \dots, n$, and a structured additive regression model with $E(y|\mathbf{x}) = h(\eta(\mathbf{x}))$, where η is an additive predictor of the form

$$\eta(\mathbf{x}) = \beta_0 + \sum_{j=1}^J f_j(\mathbf{x}).$$

- With boosting, the log likelihood of the model is maximized in a stagewise fashion via gradient ascent.
 - In each iteration, the gradient is estimated with the help of a **base-learner** regressing the gradient to the covariates.
 - Component-wise** boosting: In each iteration, only the best-fitting covariate or modeling alternative is selected to estimate the gradient.
 - ⇒ Variable selection + model choice is carried out in each boosting iteration.
 - To prevent overfitting of the data, the algorithm is stopped early (at “optimal” iteration \hat{m}_{stop} which is determined by cross-validation).
- ⇒ Interpretation of results is similar to classical ML estimation, black-box predictions can be avoided
- ⇒ Regularization via variable selection and shrinkage

Selection Bias - Typical Examples

- Consider two **non-informative** categorical covariates X and Z with M_1 and $(M_1 + M_2)$ categories, respectively.
 - ⇒ Since Z offers more flexibility in modeling the gradient of the log likelihood, it will be preferably selected (although both X and Z are non-informative).
 - Consider a **non-informative** continuous covariate X and two competing modeling alternatives for X : (a) a linear effect, (b) a smooth P-spline effect.
 - ⇒ Since the smooth effect offers more flexibility in modeling the gradient of the log likelihood, it will be preferably selected (although both effects are zero).
- ⇒ Boosting estimates tend to become too complex because of selection bias.

Penalized Base-Learners

- Consider base-learners that can be expressed as penalized linear models with hat matrix

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{K})^{-1} \mathbf{X}^T,$$

where \mathbf{X} is a suitable design matrix for some covariate(s).

- Examples of penalized base-learners:

- Ridge-penalized base-learners, for unordered categorical covariates
- Base-learners with first-order difference penalty, for ordered categorical covariates
- P-spline base-learners with a second-order difference penalty, for continuous covariates (→ decompose the smooth base-learner into an unpenalized linear function and a penalized smooth deviation from this linear function)

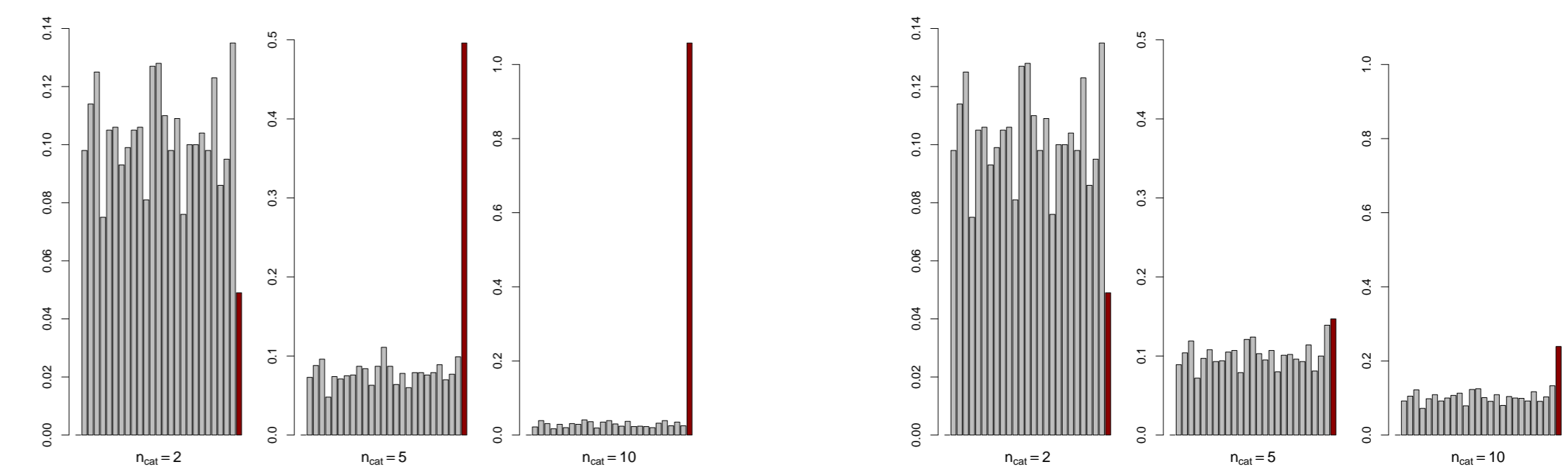
- ⇒ Base-learners can be made comparable by using the same number of df's for each penalized base-learner.
- ⇒ By theoretical considerations, define $\text{df} = \text{tr}(2\mathbf{H} - \mathbf{H}^T \mathbf{H})$.
- ⇒ Central idea: Set $\text{df} = 1$ for all base-learners to prevent selection bias.

Conclusion and Further Aspects

- Selection bias can be severely reduced if the degrees of freedom of base-learners are chosen to be equal
- It is essential to use the “correct” definition of the df's: $\text{df} = \text{tr}(2\mathbf{H} - \mathbf{H}^T \mathbf{H})$
- Correction of selection bias is computationally inexpensive:
 - Computation of smoothing parameters is only required once (at the beginning of the boosting algorithm)
 - Multidimensional cross-validation for smoothing parameters can be avoided

Simulation Example 1 - “Null Model”

- Consider a standard normally distributed outcome variable and 25 non-informative independent continuous covariates $\sim U[0, 1]$
- Add a non-informative categorical covariate Z_1 with varying numbers of categories (denoted by n_{cat})
- Use simple linear models as base-learners for each covariate
- 150 independent observations, 1000 simulation runs



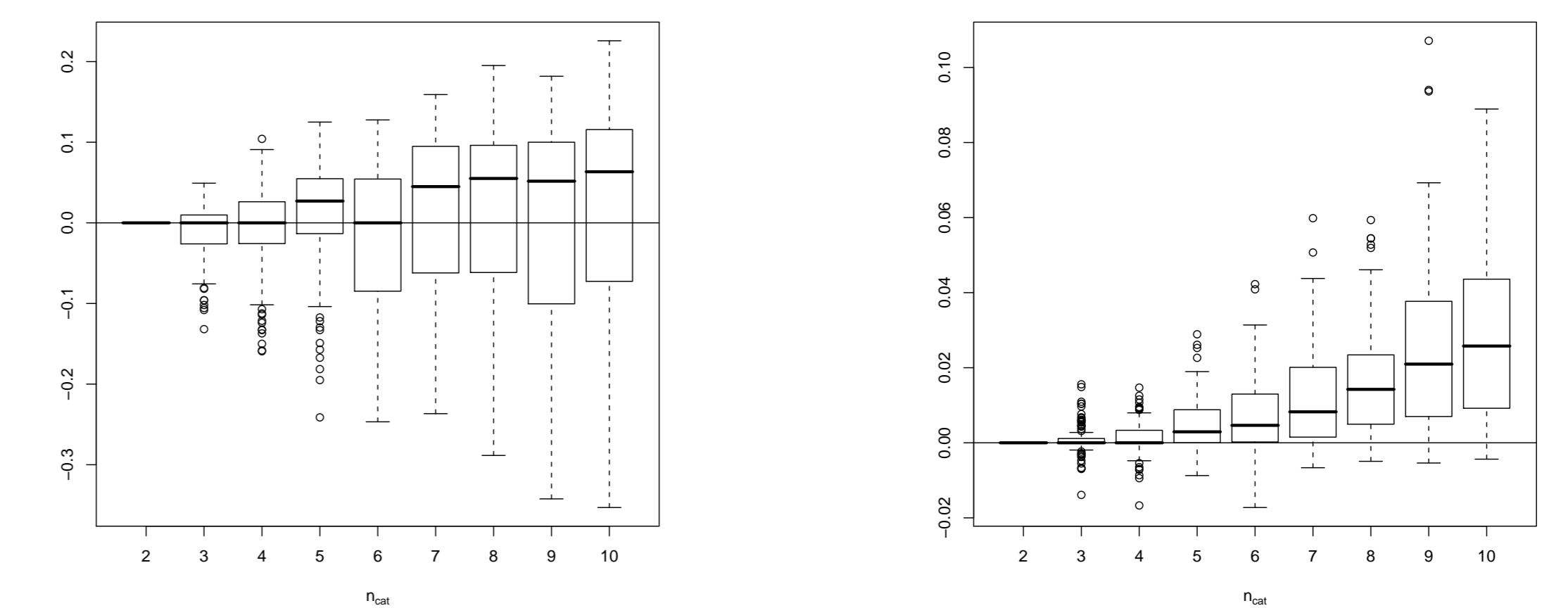
(a) Unpenalized base-learner

(b) Penalized base-learner

Average selection frequencies of covariates at iteration \hat{m}_{stop} (last bars correspond to Z_1)

Simulation Example 2 - “Power case” with Non-Informative Categorical Effect

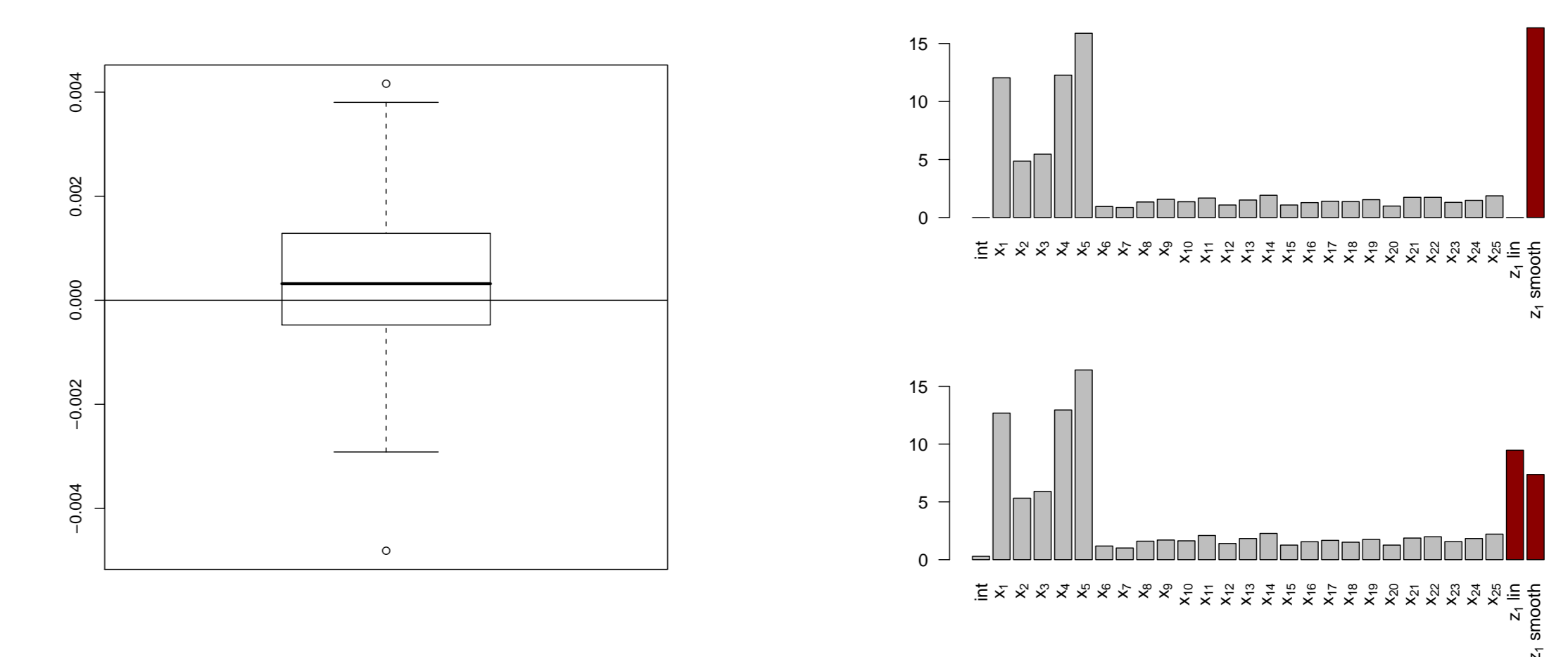
- Same as null model but with 5 informative continuous covariates (linear effects, $\beta_{\text{info}} = (-2, -1, 1, 2, 3)^T$, $R^2 = 0.3$) and 20 non-informative continuous covariates instead of 25 non-informative covariates
- 150 independent observations, 100 simulation runs
- Use MSE to evaluate variable selection bias: $\text{MSE} = \sum_{j=1}^{\tilde{p}} (\hat{\beta}_j - \beta_j)^2 / \tilde{p}$, where \tilde{p} is the number of model parameters



Differences of relative selection frequencies for Z_1 (unpenalized model - penalized model, left) and differences of MSE values (unpenalized model - penalized model, right)

Simulation Example 3 - Model with P-Spline Base-Learner

- Same as power case but with additional informative continuous covariate Z_1 instead of categorical covariate (→ additional linear effect, $\beta_{Z_1} = 1.5$, $R^2 = 0.3$)
- For the additional covariate we use a linear and a smooth modeling alternative (simple linear model vs. P-spline with varying df's)
- Use L_2 norm to measure deviation between estimated and true function estimates: $\Delta_{L_2} = \int [\hat{f}(x) - f(x)]^2 dx$



Differences between L_2 norms (model with 4 df - model with 1 df, left) and average selection frequencies for model with 4 df (upper right) and model with 1 df (lower right)

- Implementation: R package **mboost** (Hothorn et al. 2009)

Hothorn, T., P. Buhlmann, T. Kneib, M. Schmid and B. Hofner (2009): *mboost: Model-Based Boosting*. R package, version 1.1-2.
<http://cran.r-project.org/web/packages/mboost>.