mboost - Componentwise Boosting for Generalised Regression Models

Thomas Kneib & Torsten Hothorn

Department of Statistics
Ludwig-Maximilians-University Munich

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

13.8.2008

Boosting in a Nutshell

- Boosting is a simple but versatile iterative stepwise gradient descent algorithm.
- Versatility: Estimation problems are described in terms of a loss function ρ (e.g. the negative log-likelihood).
- Simplicity: Estimation reduces to iterative fitting of base-learners to residuals (e.g. regression trees).
- Componentwise boosting yields
  - a structured model fit (interpretable results),
  - model choice and variable selection.

- Example: Estimation of a generalised linear model

$$E(y|\eta) = h(\eta), \qquad \eta = \beta_0 + x_1\beta_1 + \ldots + x_p\beta_p.$$

- Employ the negative log-likelihood as the loss function ρ.

- Componentwise boosting algorithm:

(i) Initialise the parameters (e.g. $\hat{\beta}_j \equiv 0$); set $m = 0$.

(ii) Compute the negative gradients ('residuals')

$$u_i = -\frac{\partial}{\partial \eta}\rho(y_i, \eta)\bigg|_{\eta=\hat{\eta}^{[m-1]}}, \; i = 1, \ldots, n.$$

(iii) Fit least-squares base-learning procedures for all the parameters yielding

$$b_j = (X_j' X_j)^{-1} X_j' u$$

and find the best-fitting one:

$$j^* = \underset{1 \leq j \leq p}{\operatorname{argmin}} \sum_{i=1}^{n} (u_i - x_{ij} b_j)^2.$$

(iv) Update the estimates via

$$\hat{\beta}_{j^*}^{[m]} = \hat{\beta}_{j^*}^{[m-1]} + \nu b_{j^*},$$

and

$$\hat{\beta}_{j}^{[m]} = \hat{\beta}_{j}^{[m-1]} \quad \text{for all } j \neq j^*.$$

(v) If $m < m_{\text{stop}}$, increase m by 1 and go back to step (ii).

- The reduction factor ν turns the base-learner into a weak learning procedure (avoids to large steps along the gradient in the boosting algorithm).

- The componentwise strategy yields a structured model fit (recurs to single regression coefficients).

- Most crucial point: Determine optimal stopping iteration m_{stop}.

- Most frequent strategies: AIC-reduction or cross-validation.

- When stopping the algorithm, redundant covariate effects will never have been selected as the best-fitting component

  $\Rightarrow$ These drop completely out of the model.

- Componentwise boosting with early stopping implements model choice and variable selection.

mboost

- **mboost** implements a variety of base-learners and boosting algorithms for generalised regression models.

- Examples of loss functions: L_2, L_1, exponential family log-likelihoods, Huber, etc.

- Three model types:
  - `glmboost` for models with linear predictor.
  - `blackboost` for prediction oriented black-box models.
  - `gamboost` for models with additive predictors.

- Various baselearning procedures:
  - `bbs`: penalized B-splines for univariate smoothing and varying coefficients.
  - `bspatial`: penalized tensor product splines for spatial effects and interaction surfaces.
  - `brandom`: ridge regression for random intercepts and slopes.
  - `btree`: stumps for one or two variables.
  - further univariate smoothing baselearners: `bss`, `bns`.

Penalised Least Squares Base-Learners

- Several of **mboost**'s baselearning procedures are based on penalised least-squares fits.

- Characterised by the hat matrix

$$S_\lambda = X(X'X + \lambda K)^{-1}X'$$

with smoothing parameter λ and penalty matrix K.

- Crucial: Choose the smoothing parameter appropriately.

- To avoid biased selection towards more flexible effects, all base-learners should be assigned comparable degrees of freedom

$$\mathrm{df}(\lambda) = \mathrm{trace}(X(X'X + \lambda K)^{-1}X').$$

- In many cases, a reparameterisation is required to achieve suitable values for the degrees of freedom.

- Example: A linear effect remains unpenalised with penalised spline smoothing and second derivative penalty
$$\Rightarrow \qquad \mathrm{df}(\lambda) \geq 2.$$

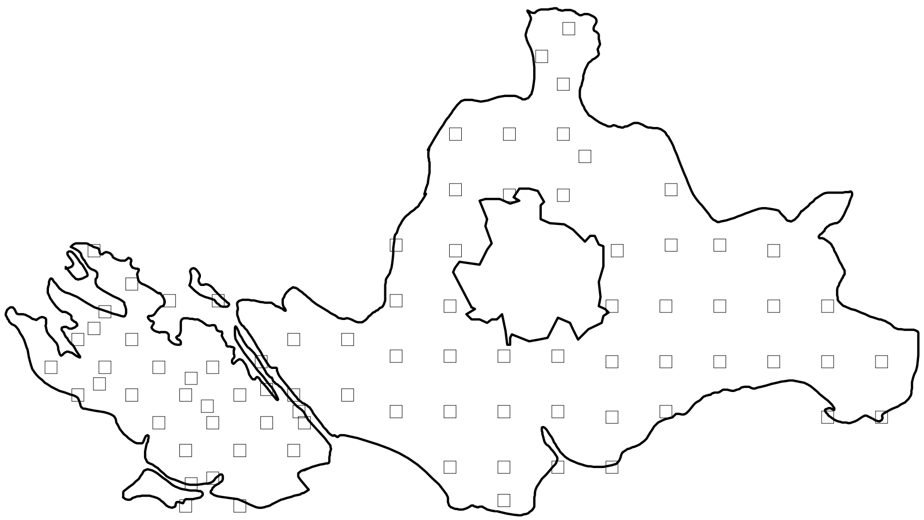
- Decompose $f(x)$ into a linear component and the deviation from the linear component.

- Assign separate base-learners (with $\mathrm{df} = 1$) to the linear effect and the deviation.

- Additional advantage: Allows to decide whether a non-linear effect is required.

Forest Health Example: Geoadditive Regression

- Aim of the study: Identify factors influencing the health status of trees.
- Database: Yearly visual forest health inventories carried out from 1983 to 2004 in a northern Bavarian forest district.
- 83 observation plots of beeches within a 15 km times 10 km area.
- Response: binary defoliation indicator y_{it} of plot i in year t (1 = defoliation higher than 25%).
- Spatially structured longitudinal data.



- Covariates:

Continuous:	average age of trees at the observation plot
	elevation above sea level in meters
	inclination of slope in percent
	depth of soil layer in centimeters
	pH-value in 0 – 2cm depth
	density of forest canopy in percent
Categorical	thickness of humus layer in 5 ordered categories
	base saturation in 4 ordered categories
Binary	type of stand
	application of fertilisation

- Specification of a logit model

$$P(y_{it} = 1) = \frac{\exp(\eta_{it})}{1 + \exp(\eta_{it})}$$

  with geoadditive predictor η_{it}.

- All continuous covariates are included with penalised spline base-learners decomposed into a linear component and the orthogonal deviation, i.e.

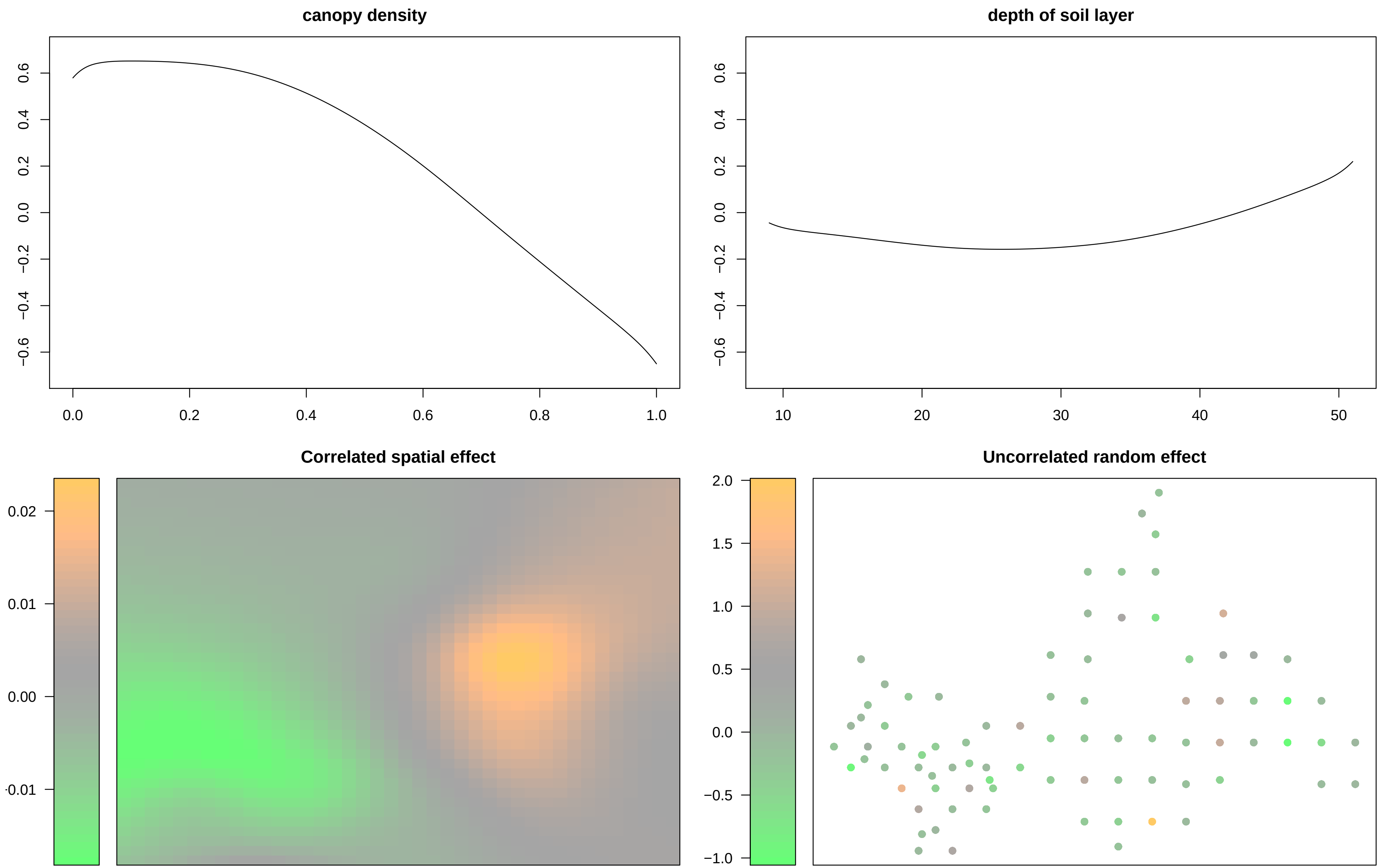
$$g(x) = x\beta + g_{\text{centered}}(x).$$

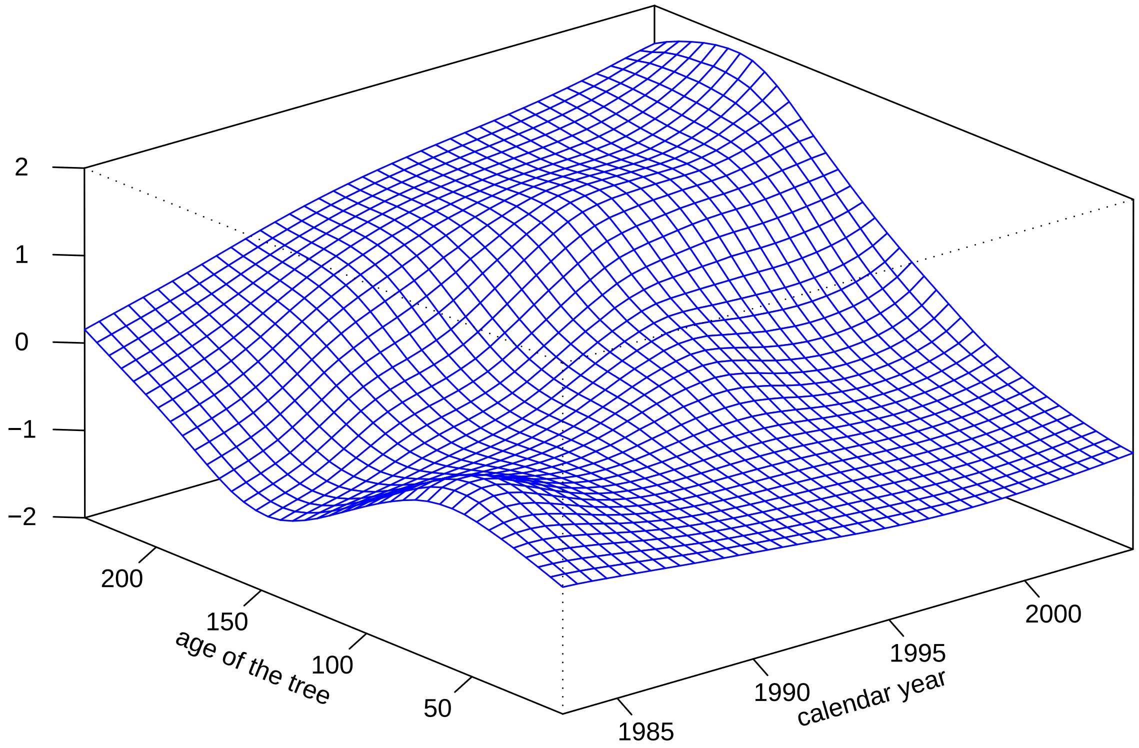
- An interaction effect between age and calendar time is included in addition (centered around the constant effect).

- The spatial effect is included both as a plot-specific random intercept and a bivariate surface of the coordinates (centered around the constant effect).

- Categorical and binary covariates are included as least-squares base-learners.

- Results:
  - No effects of ph-value, inclination of slope and elevation above sea level.
  - Parametric effects for type of stand, fertilisation, thickness of humus layer, and base saturation.
  - Nonparametric effects for canopy density and soil depth.
  - Both spatially structured effects (surface) and unstructured effect (random effect) with a clear domination of the latter.
  - Interaction effect between age and calendar time.


canopy density
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0.0
0.2
0.4
0.6
0.8
1.0
depth of soil layer
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
10
20
30
40
50
Correlated spatial effect
0.02
0.01
0.00
-0.01
Uncorrelated random effect
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0


2
1
0
−1
−2
200
150
100
50
age of the tree
1985
1990
1995
2000
calendar year

Summary

- Boosting provides both a structured model fit and a possibility for model choice and variable selection in generalised regression models.

- Simple approach based on iterative fitting of negative gradients.

- Flexible class of base-learners based on penalised least squares.

- Implemented in the R package **mboost** (Hothorn & Bühlmann with contributions by Kneib & Schmid).

- References:

  - Kneib, T., Hothorn, T. and Tutz, G. (2008): Model Choice and Variable Selection in Geoadditive Regression. To appear in *Biometrics*.
  - Bühlmann, P. and Hothorn, T. (2007): Boosting Algorithms: Regularization, Prediction and Model Fitting. *Statistical Science*, 22, 477–505.

- Find out more:

`http://www.stat.uni-muenchen.de/~kneib`