Model-based boosting in R

Introduction to Gradient Boosting

Matthias Schmid

Institut für Medizininformatik, Biometrie und Epidemiologie (IMBE)

Friedrich-Alexander-Universität Erlangen-Nürnberg

Statistical Computing 2011

Aims and scope

Why boosting?

Definition and Properties of Gradient boosting

References

Aims and scope

- Consider a sample containing the values of a response variable $\boldsymbol{Y}$ and the values of some predictor variables $\boldsymbol{X} = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_p)^\top$
- Aim: Find the "optimal" function $f^*(\boldsymbol{X})$ to predict $\boldsymbol{Y}$
- $f^*(\boldsymbol{X})$ should have a "nice" structure, for example,

$$\begin{aligned} f^*(\boldsymbol{X}) &= \beta_0 + \beta_1 \boldsymbol{X}_1 + \dots + \beta_p \boldsymbol{X}_p \quad (GLM) \quad \text{or} \\ f^*(\boldsymbol{X}) &= \beta_0 + f_1(\boldsymbol{X}_1) + \dots + f_p(\boldsymbol{X}_p) \quad (GAM) \end{aligned}$$

$\Rightarrow$ f^* should be interpretable

Example 1 - Birth weight data

- Prediction of birth weight by means of ultrasound measures (Schild et al. 2008)
  - Outcome: birth weight (BW) in g
  - Predictor variables:
    - abdominal volume (volABDO)
    - biparietal diameter (BPD)
    - head circumference (HC)
    - other predictors (measured one week before delivery)
  - Data from $n = 150$ children with birth weight $\leq 1600g$

$\Rightarrow$ Find f^* to predict BW

Birth weight data (2)

- Idea: Use 3D ultrasound measurements (left) in addition to conventional 2D ultrasound measurements (right)

Sources: www.yourultrasound.com, www.fetalultrasoundutah.com

$\Rightarrow$ Improve established formulas for weight prediction

Example 2 - Breast cancer data

- Data collected by the Netherlands Cancer Institute (van de Vijver et al. 2002)
  - 295 female patients younger than 53 years
  - Outcome: time to death after surgery (in years)
  - Predictor variables: microarray data (4919 genes) + 9 clinical variables (age, tumor diameter, ...)

$\Rightarrow$ Select a small set of marker genes ("sparse model")

$\Rightarrow$ Use clinical variables and marker genes to predict survival

Classical modeling approaches

- Classical approach to obtain predictions from birth weight data and breast cancer data: Fit additive regression models (Gaussian regression, Cox regression) using maximum likelihood (ML) estimation
- Example: Additive Gaussian model with smooth effects (represented by P-splines) for birth weight data
  $\Rightarrow f^*(\boldsymbol{X}) = \beta_0 + f_1(\boldsymbol{X}_1) + \cdots + f_p(\boldsymbol{X}_p)$

Problems with ML estimation

- ▶ Predictor variables are highly correlated

⇒ Variable selection is of interest because of multicollinearity ("Do we really need 9 highly correlated predictor variables?")

- ▶ In case of the breast cancer data: Maximum (partial) likelihood estimates for Cox regression do not exist (there are 4928 predictor variables but only 295 observations, $p \gg n$)

⇒ Variable selection because of extreme multicollinearity

⇒ We want to have a sparse (interpretable) model including the relevant predictor variables only

- ▶ Conventional methods for variable selection (univariate, forward, backward, etc.) are known to be instable and/or require the model to be fitted multiple times.

Boosting - General properties

- Gradient boosting (boosting for short) is a fitting method to minimize general types of risk functions w.r.t. a prediction function f
- Examples of risk functions: Squared error loss in Gaussian regression, negative log likelihood loss
- Boosting generally results in an *additive* prediction function, i.e., $f^*(\boldsymbol{X}) = \beta_0 + f_1(\boldsymbol{X}_1) + \cdots + f_p(\boldsymbol{X}_p)$

$\Rightarrow$ Prediction function is interpretable

$\Rightarrow$ If run until convergence, boosting can be regarded as an alternative to conventional fitting methods (Fisher scoring, backfitting) for generalized additive models.

Why boosting?

In contrast to conventional fitting methods, ...

... boosting is applicable to many different risk functions (absolute loss, quantile regression)

... boosting can be used to carry out variable selection *during the fitting process*
$\Rightarrow$ No separation of model fitting and variable selection

... boosting is applicable even if $p \gg n$

... boosting addresses multicollinearity problems (by shrinking effect estimates towards zero)

... boosting optimizes prediction accuracy (w.r.t. the risk function)

Gradient boosting - estimation problem

- Consider a one-dimensional response variable $\boldsymbol{Y}$ and a p-dimensional set of predictors $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_p)^\top$
- Aim: Estimation of

$$f^* := \underset{f(\cdot)}{\operatorname{argmin}}\, \mathsf{E}[\rho(\boldsymbol{Y}, f(\boldsymbol{X}))] \ ,$$

  where ρ is a loss function that is assumed to be differentiable with respect to a prediction function $f(\boldsymbol{X})$
- Examples of loss functions:
  - $\rho = (\boldsymbol{Y} - f(\boldsymbol{X}))^2 \rightarrow$ squared error loss in Gaussian regression
  - Negative log likelihood function of a statistical model

Gradient boosting - estimation problem (2)

- In practice, we usually have a set of realizations $X = (X_1, \dots, X_n)$, $Y = (Y_1, \dots, Y_n)$ of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively

⇒ Minimization of the empirical risk

$$\mathcal{R} = \frac{1}{n} \sum_{i=1}^{n} \rho(Y_i, f(X_i))$$

with respect to f

- Example: $\mathcal{R} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - f(X_i))^2$ corresponds to minimizing the expected squared error loss

Naive functional gradient descent (FGD)

- Idea: use gradient descent methods to minimize $\mathcal{R} = \mathcal{R}(f_{(1)}, \ldots, f_{(n)})$ w.r.t. $f_{(1)} = f(X_1), \ldots, f_{(n)} = f(X_n)$
- Start with offset values $\hat{f}^{[0]}_{(1)}, \ldots, \hat{f}^{[0]}_{(n)}$
- In iteration m:

$$\begin{pmatrix} \hat{f}^{[m]}_{(1)} \\ \vdots \\ \hat{f}^{[m]}_{(n)} \end{pmatrix} = \begin{pmatrix} \hat{f}^{[m-1]}_{(1)} \\ \vdots \\ \hat{f}^{[m-1]}_{(n)} \end{pmatrix} + \nu \cdot \begin{pmatrix} -\frac{\partial \mathcal{R}}{\partial f_{(1)}}(\hat{f}^{[m-1]}_{(1)}) \\ \vdots \\ -\frac{\partial \mathcal{R}}{\partial f_{(n)}}(\hat{f}^{[m-1]}_{(n)}) \end{pmatrix},$$

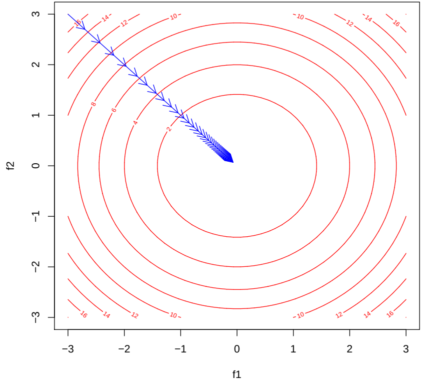
  where ν is a step length factor

$\Rightarrow$ Principle of *steepest descent*

Naive functional gradient descent (2)

(Very) simple example: $n = 2$, $Y_1 = Y_2 = 0$, $\rho =$ squared error loss

$\Rightarrow \mathcal{R} = \frac{1}{2}\left(f_{(1)}^2 + f_{(2)}^2\right)$



Naive functional gradient descent (3)

- Increase m until the algorithm converges to some values $\hat{f}_{(1)}^{[m_{\text{stop}}]}, \ldots, \hat{f}_{(n)}^{[m_{\text{stop}}]}$
- Problem with **naive** gradient descent:
  - No predictor variables involved
  - Structural relationship between $\hat{f}_{(1)}^{[m_{\text{stop}}]}, \ldots, \hat{f}_{(n)}^{[m_{\text{stop}}]}$ is ignored ($\hat{f}_{(1)}^{[m]} \to Y_1, \ldots, \hat{f}_{(n)}^{[m]} \to Y_n$)
  - "Predictions" only for observed values $Y_1, \ldots, Y_n$

Gradient Boosting

- Solution: Estimate the negative gradient in each iteration
- Estimation is performed by some base-learning procedure regressing the negative gradient on the predictor variables

  $\Rightarrow$ base-learning procedure ensures that $\hat{f}_{(1)}^{[m_{\text{stop}}]}, \ldots, \hat{f}_{(n)}^{[m_{\text{stop}}]}$ are predictions from a statistical model depending on the predictor variables
- To do this, we specify a set of regression models ("base-learners") with the negative gradient as the dependent variable
- In many applications, the set of base-learners will consist of p simple regression models ($\Rightarrow$ one base-learner for each of the p predictor variables, "component-wise gradient boosting")

Gradient Boosting (2)

Functional gradient descent (FGD) boosting algorithm:

1. Initialize the n-dimensional vector $\hat{f}^{[0]}$ with some offset values (e.g., use a vector of zeroes). Set $m = 0$ and specify the set of base-learners. Denote the number of base-learners by P.

2. Increase m by 1. Compute the negative gradient $-\frac{\partial}{\partial f}\rho(Y, f)$ and evaluate at $\hat{f}^{[m-1]}(X_i)$, $i = 1, \dots, n$. This yields the negative gradient vector

$$U^{[m-1]} = (U_i^{[m-1]})_{i=1,\dots,n} := \left(-\frac{\partial}{\partial f}\rho(Y, f)\Big|_{Y=Y_i, f=\hat{f}^{[m-1]}(X_i)}\right)_{i=1,\dots,n}$$

⋮

Gradient Boosting (3)

⋮

3. Estimate the negative gradient $U^{[m-1]}$ by using the base-learners (i.e., the P regression estimators) specified in Step 1.

   This yields P vectors, where each vector is an estimate of the negative gradient vector $U^{[m-1]}$.

   Select the base-learner that fits $U^{[m-1]}$ best ($\rightarrow$ min. SSE). Set $\hat{U}^{[m-1]}$ equal to the fitted values from the corresponding best model.

⋮

Gradient Boosting (4)

⋮

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

5. Iterate Steps 2 - 4 until $m = m_{\text{stop}}$.

- The step length factor ν could be chosen adaptively. Usually, an adaptive strategy does not improve the estimates of f^* but will only lead to an increase in running time
  $\Rightarrow$ choose ν small ($\nu = 0.1$) but fixed

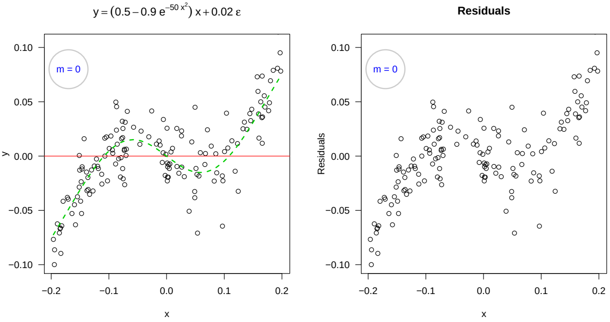
Schematic overview of Step 3 in iteration m

- Component-wise gradient boosting with one base-learner for each predictor variable:

$$
\begin{array}{ccccc}
U^{[m-1]} & \sim & \boldsymbol{X}_1 & & \\
U^{[m-1]} & \sim & \boldsymbol{X}_2 & & \\
 & \vdots & & & \\
\boxed{U^{[m-1]} \sim \boldsymbol{X}_j} & & & \xleftrightarrow{\text{best-fitting base-learner}} & \hat{U}^{[m-1]} \\
 & \vdots & & & \\
U^{[m-1]} & \sim & \boldsymbol{X}_p & &
\end{array}
$$

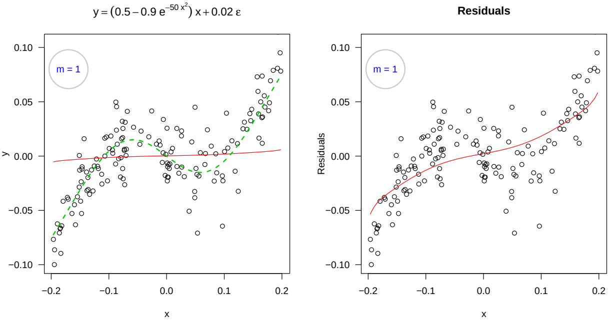
Simple example

- In case of Gaussian regression, gradient boosting is equivalent to iteratively re-fitting the residuals of the model.



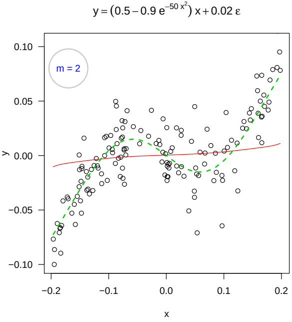
Simple example

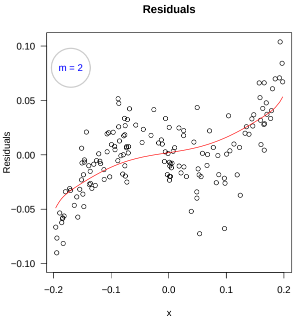
- In case of Gaussian regression, gradient boosting is equivalent to iteratively re-fitting the residuals of the model.



Simple example

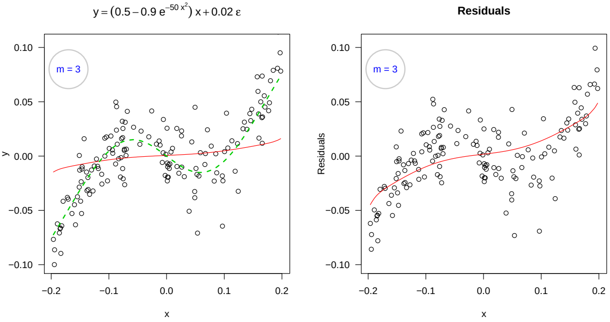
- In case of Gaussian regression, gradient boosting is equivalent to iteratively re-fitting the residuals of the model.





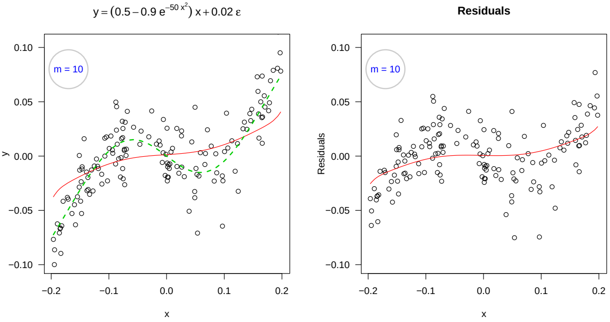
Simple example

- In case of Gaussian regression, gradient boosting is equivalent to iteratively re-fitting the residuals of the model.



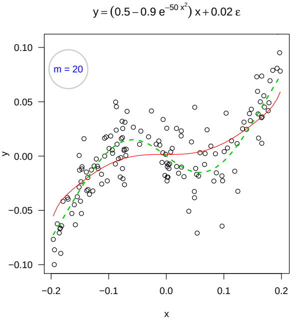
Simple example

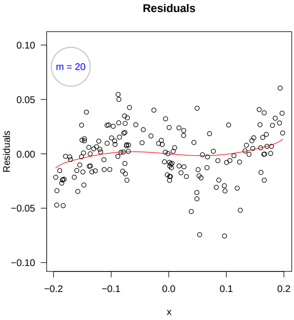
- In case of Gaussian regression, gradient boosting is equivalent to iteratively re-fitting the residuals of the model.



Simple example

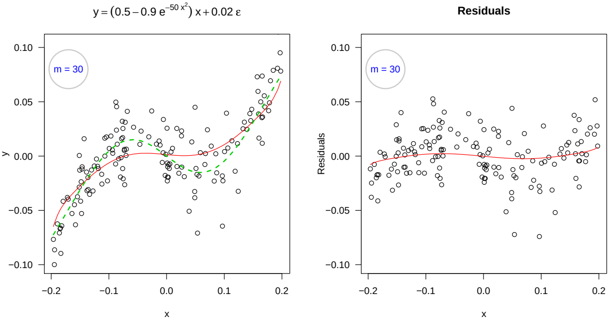
- In case of Gaussian regression, gradient boosting is equivalent to iteratively re-fitting the residuals of the model.





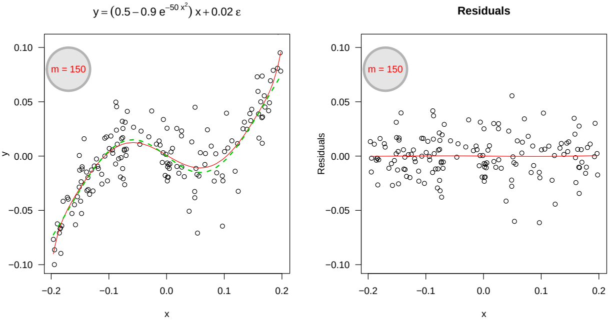
Simple example

- In case of Gaussian regression, gradient boosting is equivalent to iteratively re-fitting the residuals of the model.



Simple example

- In case of Gaussian regression, gradient boosting is equivalent to iteratively re-fitting the residuals of the model.



Properties of gradient boosting

- It is clear from Step 4 that the predictions of $Y_1, \ldots, Y_n$ in iteration m_{stop} take the form of an additive function:

$$\hat{f}^{[m_{\text{stop}}]} = \hat{f}^{[0]} + \nu\,\hat{U}^{[0]} + \cdots + \nu\,\hat{U}^{[m_{\text{stop}}-1]}$$

- The structure of the prediction function depends on the choice of the base-learners
  - For example, linear base-learners result in linear prediction functions
  - Smooth base-learners result in additive prediction functions with smooth components

$\Rightarrow$ $\hat{f}^{[m_{\text{stop}}]}$ has a meaningful interpretation

Gradient boosting with early stopping

- Gradient boosting has a „built-in“ mechanism for base-learner selection in each iteration.

⇒ This mechanism will carry out variable selection.

- Gradient boosting is applicable even if $p > n$.
- In case $p > n$, it is usually desirable to select a small number of informative predictor variables ("sparse solution").
- If $m \rightarrow \infty$, the algorithm will select non-informative predictor variables.

  ⇒ Overfitting can be avoided if the algorithm is *stopped early*, i.e., if m_{stop} is considered as a tuning parameter of the algorithm

Illustration of variable selection and early stopping

- Very simple example: 3 predictor variables $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{X}_3$,
  3 linear base-learners with coefficient estimates $\hat{\beta}_j^{[m]}$, $j = 1, 2, 3$
- Assume that $m_{\text{stop}} = 5$
- Assume that $\boldsymbol{X}_1$ was selected in the first, second and fifth iteration
- Assume that $\boldsymbol{X}_3$ was selected in the third and forth iteration

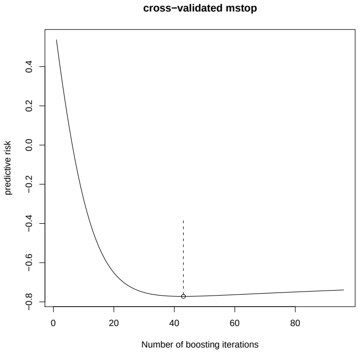
$$
\begin{aligned}
\hat{f}^{[m_{\text{stop}}]} &= \hat{f}^{[0]} + \nu\,\hat{U}^{[0]} + \nu\,\hat{U}^{[1]} + \nu\,\hat{U}^{[2]} + \nu\,\hat{U}^{[3]} + \nu\,\hat{U}^{[4]} \\
&= \hat{\beta}^{[0]} + \nu\,\hat{\beta}_1^{[0]}\,\boldsymbol{X}_1 + \nu\,\hat{\beta}_1^{[1]}\,\boldsymbol{X}_1 + \nu\,\hat{\beta}_3^{[2]}\,\boldsymbol{X}_3 + \nu\,\hat{\beta}_3^{[3]}\,\boldsymbol{X}_3 + \nu\,\hat{\beta}_1^{[4]}\,\boldsymbol{X}_1 \\
&= \hat{\beta}^{[0]} + \nu\left(\hat{\beta}_1^{[0]} + \hat{\beta}_1^{[1]} + \hat{\beta}_1^{[4]}\right)\boldsymbol{X}_1 + \nu\left(\hat{\beta}_3^{[2]} + \hat{\beta}_3^{[3]}\right)\boldsymbol{X}_3 \\
&= \hat{\beta}^{[0]} + \hat{\beta}_1^{*}\boldsymbol{X}_1 + \hat{\beta}_3^{*}\boldsymbol{X}_3
\end{aligned}
$$

$\Rightarrow$ Linear prediction function

$\Rightarrow$ $\boldsymbol{X}_2$ is not included in the model (variable selection)

How should the stopping iteration be chosen?

- Use cross-validation techniques to determine m_{stop}



$\Rightarrow$ The stopping iteration is chosen such that it *maximizes prediction accuracy*.

Shrinkage

- Early stopping will not only result in sparse solutions but will also lead to shrunken effect estimates ($\rightarrow$ only a small fraction of $\hat{U}$ is added to the estimates in each iteration).
- Shrinkage leads to a downward bias (in absolute value) but to a smaller variance of the effect estimates (similar to Lasso or Ridge regression).

$\Rightarrow$ Multicollinearity problems are addressed.

Further aspects

- There are many types of boosting methods, e.g.,
  - tree-based boosting (AdaBoost, Freund & Schapire 1997)
  - likelihood-based boosting (Tutz & Binder 2006)
- Here we consider *gradient boosting*
  - Flexible method to fit many types of statistical models in high- *and* low-dimensional settings
  - *Regularization* of estimates via variable selection and shrinkage
- Implemented in R package **mboost** (Hothorn et al. 2010, 2011)

References

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

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

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid and B. Hofner (2011): mboost: Model-Based Boosting. R package version 2.1-0. https://r-forge.r-project.org/projects/mboost/

Schild, R. L., M. Maringa, J. Siemer, B. Meurer, N. Hart, T. W. Goecke, M. Schmid, T. Hothorn and M. E. Hansmann (2008). Weight estimation by three-dimensional ultrasound imaging in the small fetus. Ultrasound in Obstetrics and Gynecology 32, 168-175.

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

van de Vijver, M. J., Y. D. He, L. J. van't Veer, H. Dai, A. A. M. Hart, D. W. Voskuil et al. (2002). A gene-expression signature as a predictor of survival in breast cancer. New England Journal of Medicine 347, 1999-2009.