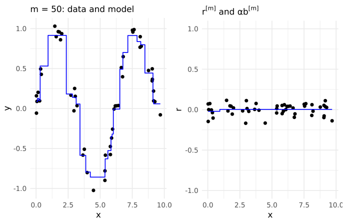
Introduction to Machine Learning

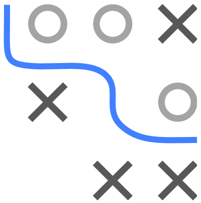
Boosting
Gradient Boosting with Trees 1



Learning goals

- Examples for GB with trees
- Understand relationship between model structure and interaction depth

GRADIENT BOOSTING WITH TREES



Trees are most popular BLs in GB.

Reminder: advantages of trees

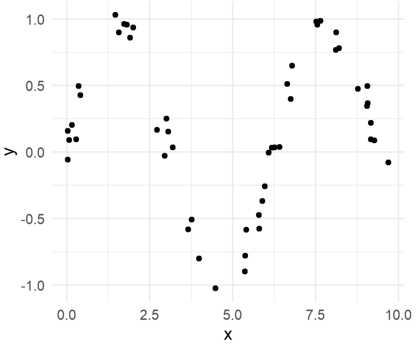
- No problems with categorical features.
- No problems with outliers in feature values.
- No problems with missing values.
- No problems with monotone transformations of features.
- Trees (and stumps!) can be fitted quickly, even for large n.
- Trees have a simple, built-in type of variable selection.

GB with trees inherits these, and strongly improves predictive power.

EXAMPLE 1

Simulation setting:

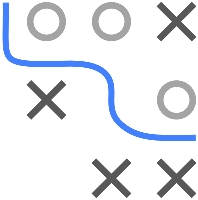
- Given: one feature x and one numeric target variable y of 50 observations.
- x is uniformly distributed between 0 and 10.
- y depends on x as follows: $y^{(i)} = \sin\left(x^{(i)}\right) + \epsilon^{(i)}$ with $\epsilon^{(i)} \sim \mathcal{N}(0, 0.01)$, $\forall i \in \{1, \ldots, 50\}$.



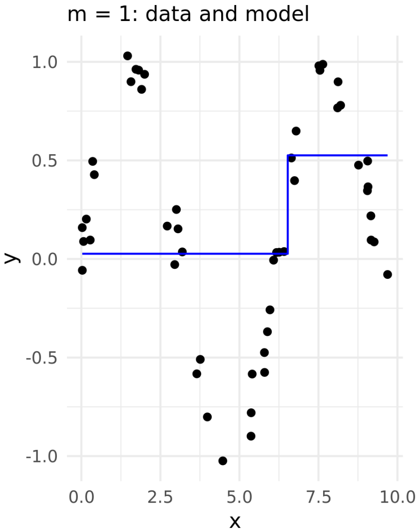
Aim: we want to fit a gradient boosting model to the data by using stumps as base learners.

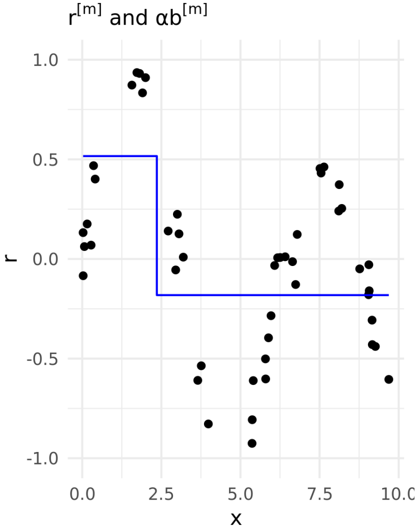
Since we are facing a regression problem, we use $L2$ loss.

EXAMPLE 1



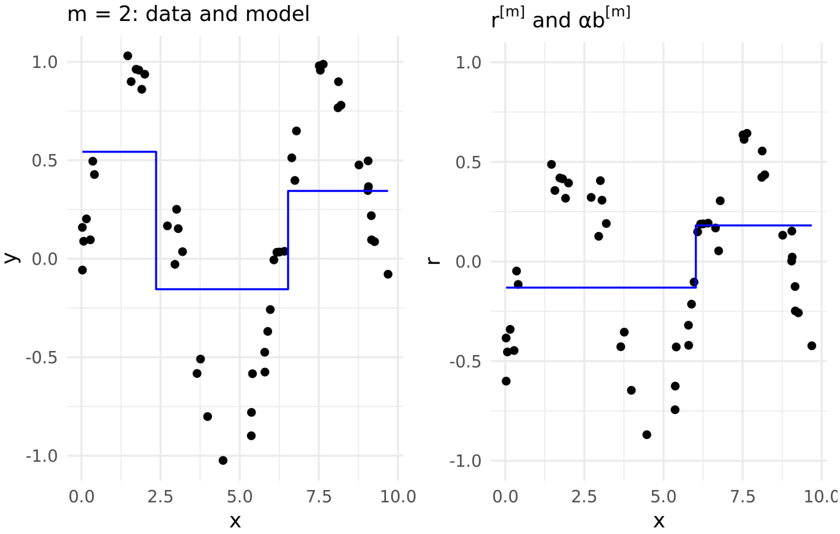
Repeat step (1) to (3):





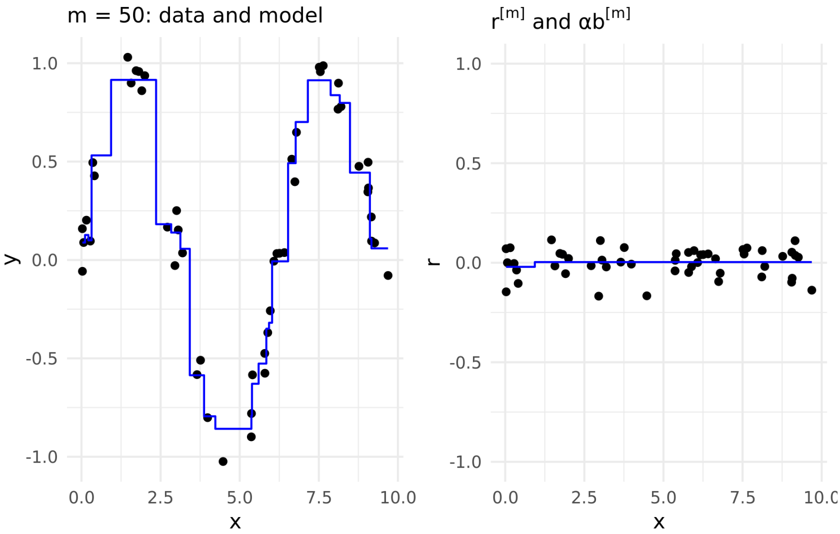
EXAMPLE 1

Repeat step (1) to (3):



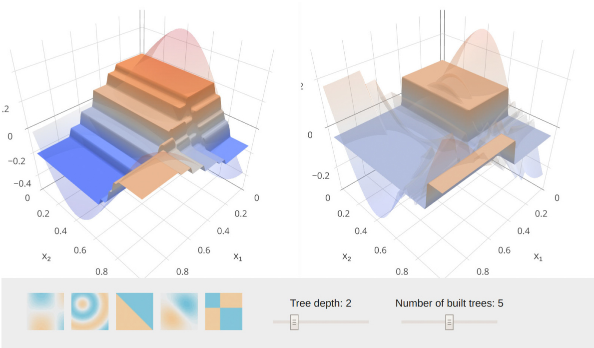
EXAMPLE 1

Repeat step (1) to (3):



EXAMPLE 2

This website shows on various 3D examples how tree depth and number of iterations influence the model fit of a GBM with trees.



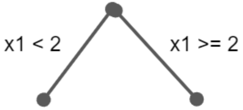
MODEL STRUCTURE AND INTERACTION DEPTH

Model structure directly influenced by depth of $b^{[m]}(\mathbf{x})$.

$$f(\mathbf{x}) = \sum_{m=1}^{M} \alpha^{[m]} b^{[m]}(\mathbf{x})$$

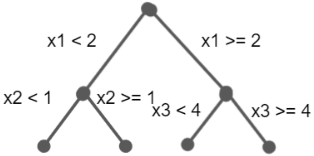
Remember how we can write trees as additive model over paths to leafs.

With stumps (depth = 1), $f(\mathbf{x})$ is additive model (GAM) without interactions:



$$f(\mathbf{x}) = f_0 + \sum_{j=1}^{p} f_j(x_j)$$

With trees of depth 2, we have two-way interactions:



$$f(\mathbf{x}) = f_0 + \sum_{j=1}^{p} f_j(x_j) + \sum_{j \neq k} f_{j,k}(x_j, x_k)$$

with f_0 being a constant intercept.