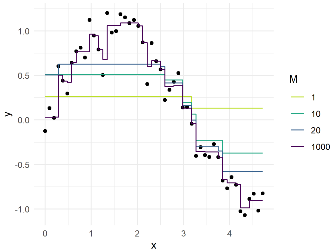
Introduction to Machine Learning

Gradient Boosting: Regularization



Learning goals

- Learn about three main regularization options: number of iterations, tree depth and shrinkage
- Understand how regularization influences model fit

ITERS, TREE DEPTH, LEARN RATE

GB can overfit easily, due to its aggressive loss minimization.

Options for regularization:

- Limit nr of iters *M*, i.e., additive components ("early stopping"),
- Limit depth of trees. Can also be interpreted as choosing the order of interaction (see later).
- Use a small learn rate α for only mild model updates.
  α a.k.a. shrinkage.

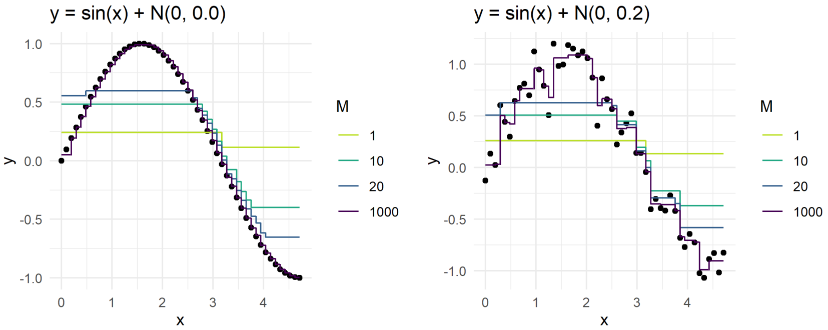
Practical hints:

- Optimal values for *M* and α strongly depend on each other: by increasing *M* one can use a smaller value for α and vice versa.
- Fast option = Make α small and choose *M* by CV.
- Probably best to tune all 3 hyperpars jointly via, e.g., CV.

STOCHASTIC GRADIENT BOOSTING

- Minor modification to incorporate the advantages of bagging
- In each iter, we only fit on a random subsample of the train data
- Especially for small train sets, this often leads helps
- Size of random sets = new hyperpar

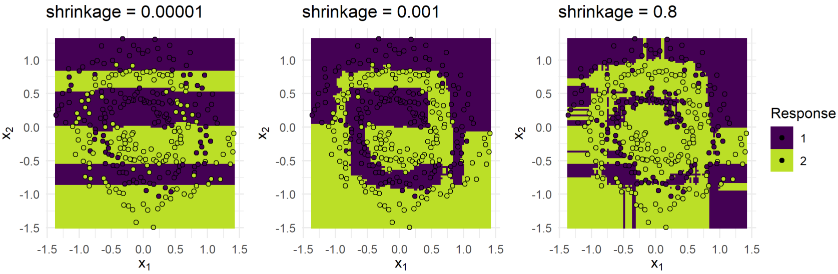
EXAMPLE: SINUSOIDAL WITH TREE STUMPS



Works quite nicely without noise, but overfits on the RHS.

EXAMPLE: SPIRALS DATA

We examine effect of learn rate, with fixed nr of trees and fixed depth.

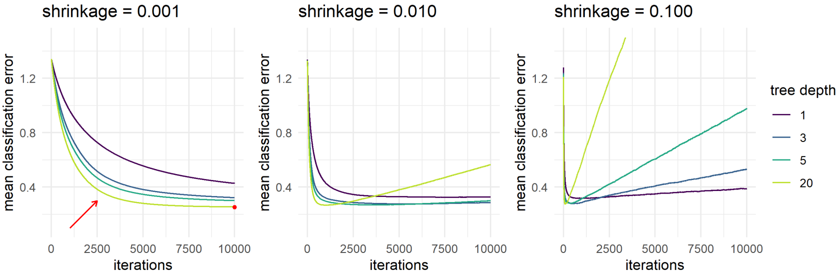


We observe an oversmoothing effect in the left scenario with strong regularization (i.e., very small learning rate) and overfitting when regularization is too weak (right). $\alpha = 0.001$ yields a pretty good fit.

EXAMPLE: SPAM DETECTION WITH TREES

Hyperpar	Range
Loss	Bernoulli (for classification)
Number of trees M	$\{0, 1, \dots, 10000\}$
Shrinkage α	$\{0.001, 0.01, 0.1\}$
Max. tree depth	$\{1, 3, 5, 20\}$

Use 3-CV in grid search; optimal config in red:


shrinkage = 0.001
shrinkage = 0.010
shrinkage = 0.100
mean classification error
iterations
tree depth