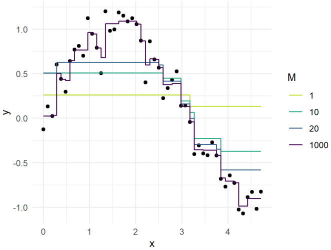
# Introduction to Machine Learning

# Gradient Boosting: Regularization



## Learning goals

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

# REGULARIZATION AND SHRINKAGE

If GB runs for a large number of iterations, it can overfit due to its aggressive loss minimization.

**Options for regularization:**

- Limit the number of boosting iterations $M$ (“early stopping”), i.e., limit the number of additive components.
- Limit the depth of the trees. This can also be interpreted as choosing the order of interaction.
- Shorten the learning rate $\beta^{[m]}$ of each iteration.

Note: If $\beta^{[m]}$ is found via line search, we multiply the next base learner by an additional constant **shrinkage parameter** $\nu \in (0, 1]$ to shorten the step length. In the case of a (commonly used) constant learning rate $\beta$, $\nu$ can be neglected by choosing a smaller value for $\beta$. Hence, on the following slides, we call $\beta$ the shrinkage parameter or learning rate.

# REGULARIZATION AND SHRINKAGE

Obviously, the optimal values for *M* and $\beta$ strongly depend on each other: by increasing *M* one can use a smaller value for $\beta$ and vice versa.

In practice, a common recommendation to find a good first model without tuning both parameters is thus to choose $\beta$ quite small and determine *M* by cross-validation.

It is probably best to tune all three parameters (*M* and $\beta$ as well as the tree depths) jointly based on the training data via cross-validation or a related method.

# STOCHASTIC GRADIENT BOOSTING

**Stochastic gradient boosting** is a minor modification to boosting to incorporate the advantages of bagging into the method. The idea was formulated quite early by Breiman.

Instead of fitting on all the data points, a random subsample is drawn in each iteration.

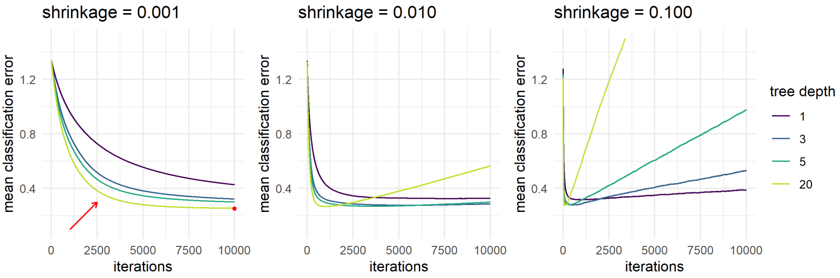
Especially for small training sets, this simple modification often leads to substantial empirical improvements. How large the improvements are depends on data structure, size of the dataset, base learner and size of the subsamples (so this is another tuning parameter).

# EXAMPLE: SPAM DETECTION

We fit a gradient boosting model for different parameter values:

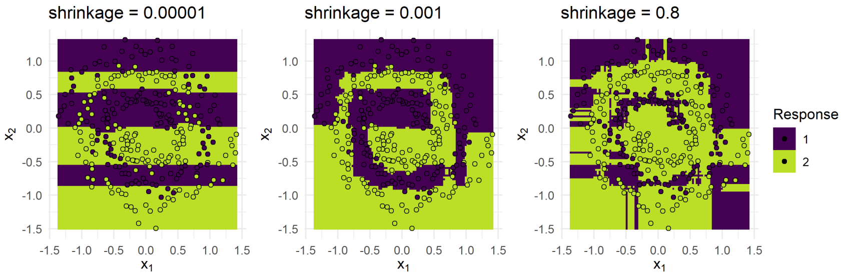
| Parameter name | Values |
|---|---|
| Loss | Bernoulli (for classification) |
| Number of trees *M* | $\{0, 1, \dots, 10000\}$ |
| Shrinkage $\beta$ | $\{0.001, 0.01, 0.1\}$ |
| Max. tree depth | $\{1, 3, 5, 20\}$ |

We consider the 3-CV test error to find the optimal parameters (red):



# EXAMPLE: SPIRALS DATA

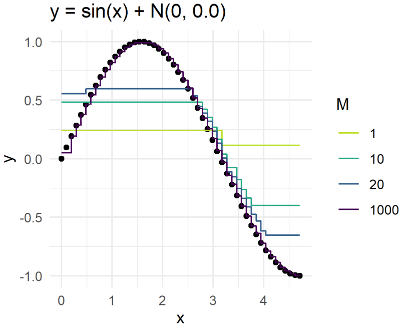
Consider the $\mathrm{spirals}$ data set with $sd = 0.1$ and $n = 300$. We examine the effect of the shrinkage parameter, holding the number of trees fixed at 10k and choosing a tree depth of 10:

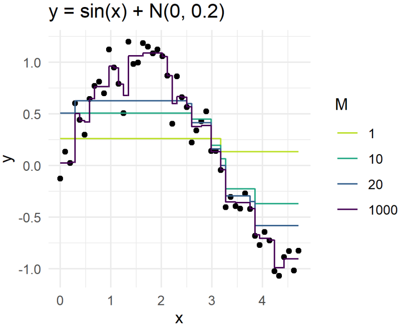


We observe an oversmoothing effect in the left scenario with strong regularization (i.e., very small learning rate) and overfitting when regularization is too weak (right). $\beta = 0.001$ yields a pretty good fit.

# EXAMPLE: SINUSOIDAL DATA

Here we choose a tree-stump base learner to model univariate data with sinusoidal behavior.


y = sin(x) + N(0, 0.0)


y = sin(x) + N(0, 0.2)

- Iterating this very simple base learner achieves a rather nice approximation of a smooth model in the end.
- Again, the model overfits the noisy case with less regularization.