Introduction to Machine Learning

CART: Stopping Criteria & Pruning

compstat-lmu.github.io/lecture_i2ml

OVERFITTING TREES

The **recursive partitioning** procedure used to grow a CART would run until every leaf only contains a single observation.

- Problem 1: This would take a very long time, as the amount of splits we have to try *grows exponentially* with the number of leaves in the trees.
- Problem 2: At some point before that we should stop splitting nodes into ever smaller child nodes: very complex trees with lots of branches and leaves will *overfit the training data.*
- Problem 3: However, it is very hard to tell where we should stop while we're growing the tree: Before we actually try all possible additional splits further down a branch, we can't know whether any one of them will be able to reduce the risk by a lot (*horizon effect*).

STOPPING CRITERIA

Problems 1 and 2 can be “solved” by defining different **stopping criteria**:

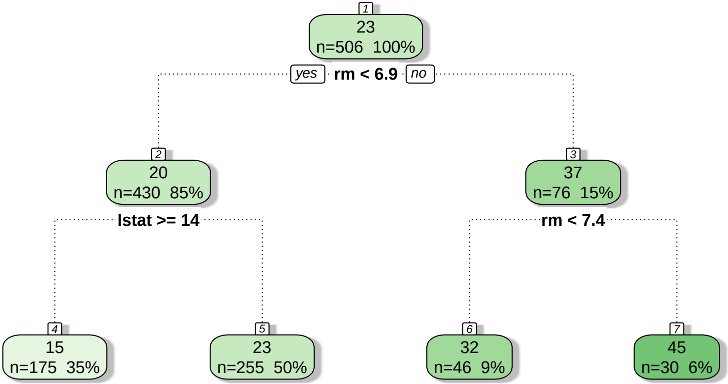
- Stop once the tree has reached a certain number of leaves.
- Don’t try to split a node further if it contains too few observations.
- Don’t perform a split that results in child nodes with too few observations.
- Don’t perform a split unless it achieves a certain minimal improvement of the empirical risk in the child nodes compared to the empirical risk in the parent node.
- Obviously: Stop once all observations in a node have the same target value (**pure node**) or identical values for all features.

PRUNING

We try to solve problem 3 by **pruning**:

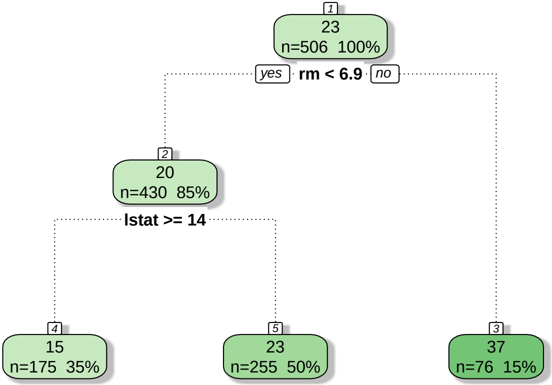
- a method to select the optimal size of a tree
- Finding a combination of suitable strict stopping criteria ("pre-pruning") is a hard problem: there are many different stopping criteria and it's hard to find the best combination (see chapter on **tuning**)
- Better: Grow a large tree, then remove branches so that the resulting smaller tree has optimal cross-validation risk
- Feasible without cross-validation: Grow a large tree, then remove branches so that the resulting smaller tree has a good balance between training set performance (risk) and complexity (i.e., number of terminal nodes). The trade-off between complexity and accuracy is governed by a **complexity parameter**.

PRUNING



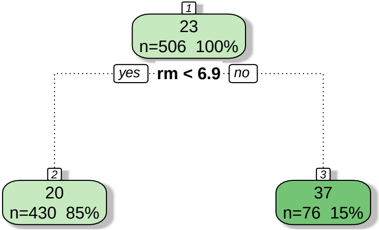
Full tree

PRUNING



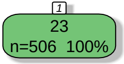
Pruning with complexity parameter = 0.072.

PRUNING



Pruning with complexity parameter = 0.171.

PRUNING



Pruning with complexity parameter = 0.453.