# Introduction to Machine Learning

## Chapter 14: Trees cont.

**Bernd Bischl, Christoph Molnar**

Department of Statistics – LMU Munich

Winter term 2017/18

# MONOTONE FEATURE TRANSFORMATIONS

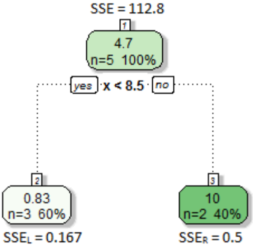
Monotone transformations of one or several features will not change the value of the impurity measure, neither the structure of the tree (just the numerical value of the split point).

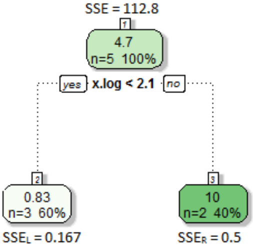
Original data

| x | 1 | 2 | 7.0 | 10 | 20 |
|---|---|---|---|---|---|
| y | 1 | 1 | 0.5 | 10 | 11 |

Data with log-transformed *x*

| log(x) | 0 | 0.7 | 1.9 | 2.3 | 3 |
|---|---|---|---|---|---|
| y | 1 | 1.0 | 0.5 | 10.0 | 11 |





# CART: STOPPING-CRITERIA

- Minimal number of observations per node, for a split to be tried
- Minimal number of observations that must be contained in a leaf
- Minimal increase in goodness of fit that must be reached for a split to be tried
- Maximum number of levels for your tree

# CART: OVERFITTING

- The CART-Algorithm could just continue until there is a single observation in each node

⇒ Complexity (and hence the danger of overfitting) increases with the number of splits / levels / leafs

# CART: CATEGORICAL PREDICTORS

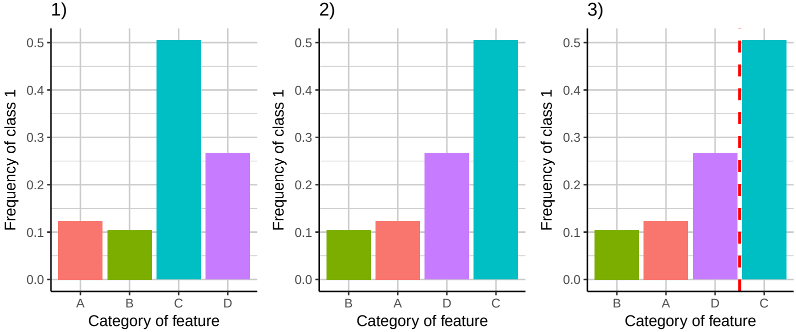
- For a nominal scaled feature with $Q$ categories, there are $2^{Q-1} - 1$ possible partitions of the $Q$ values into two groups:
  - There are $2^Q$ ways to assign $Q$ distinct values to the left or right node.
  - Two of these configurations lead to an empty node, while the other one contains all observations. Discarding these configurations leads to $2^Q - 2$ possible partitions.
  - Symmetry halves the number of possible partitions: $\frac{1}{2}(2^Q - 2) = 2^{Q-1} - 1$

  $\Rightarrow$ computations become prohibitive for large values of $Q$
- But for regression with squared loss and binary classification shortcuts exist.

# CART: CATEGORICAL PREDICTORS

For $0 - 1$ responses, in each node:

1. Calculate the proportion of 1-outcomes for each category of the feature.
2. Sort the categories according to these proportions.
3. The feature can then be treated as if it were an ordered categorical feature ($Q - 1$ possible splits).



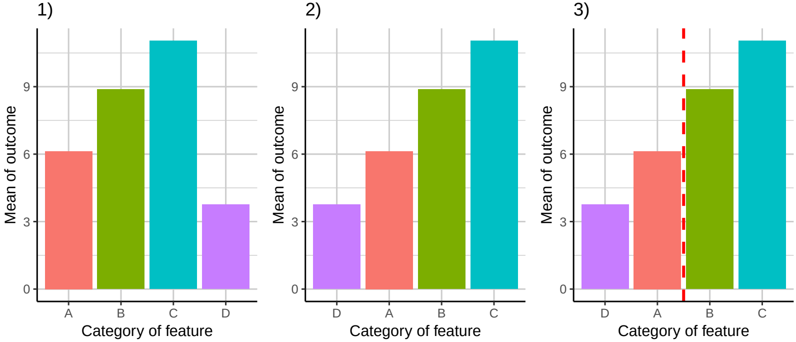
# CART: CATEGORICAL PREDICTORS

- This procedure obtains the optimal split for entropy and Gini index.
- This result also holds for regression trees (with squared error loss) – the categories are ordered by increasing mean of the outcome (see next slide).
- The proofs are not trivial and can be found here:
  - for 0-1 responses:
    - Breiman, 1984, Classification and Regression Trees.
    - Ripley, 1996, Pattern Recognition and Neural Networks.
  - for continuous responses:
    - Fisher, 1958, On grouping for maximum homogeneity.
- Such simplifications are not known for multiclass problems.

# CART: CATEGORICAL PREDICTORS

For continuous responses, in each node:

1. Calculate the mean of the outcome in each category.
2. Sort the categories by increasing mean of the outcome.
3. The feature can then be treated as if it were an ordered categorical feature ($Q - 1$ possible splits).



# CART: MISSING PREDICTOR VALUES

Two approaches:

1. Missing values of a categorical variable are treated as an own category
2. When considering a predictor for a split, only use the observations for which the predictor is not missing.
   To pass observations with missing values down the tree (during fitting or predicting), we have to find surrogate variables, that produce similar splits.

# ADVANTAGES

- Model is easy to comprehend, graphical representation
- Categorical features can easily be handled
- Missing values can be handled
- No problems with outliers in features
- Monotone transformations of features change nothing
- Interaction effects between features are easily possible
- Works for (some) non-linear functions
- Inherent feature selection
- Quite fast, scales well with larger data
- Trees are flexible by creating a custom split criterion and leaf-node prediction rule (clustering trees, semi-supervised trees, density estimation, etc.)

# DISADVANTAGES

- High instability (variance) of the trees: Small changes in the data could lead to completely different splits, thus, to completely different trees → Decisions on the upper level influence decisions on lower levels (“mistakes” in upper levels proceed to the lower ones.)
- Prediction function isn’t smooth because a step function is fitted.
- Linear dependencies must be modeled over several splits → Simple linear correlations must be translated into a complex tree structure (see the following example)
- Really not the best predictor: Combine with bagging (forest) or boosting! (This will also be illustrated in a small benchmark at the end of the random forest chapter.)

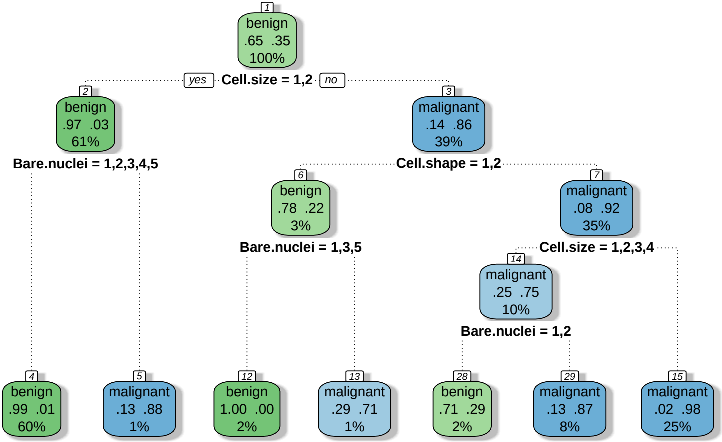
# DISADVANTAGES

High instability of trees will be demonstrated using the Wisconsin Breast Cancer data set. It has 699 observations on 9 features and one target class with values “benign” and “malignant”.

| Feature name | Explanation |
|---|---|
| `Cl.thickness` | Clump Thickness |
| `Cell.size` | Uniformity of Cell Size |
| `Cell.shape` | Uniformity of Cell Shape |
| `Marg.adhesion` | Marginal Adhesion |
| `Epith.c.size` | Single Epithelial Cell Size |
| `Bare.nuclei` | Bare Nuclei |
| `Bl.cromatin` | Bland Chromatin |
| `Normal.nucleoli` | Normal Nucleoli |
| `Mitoses` | Mitoses |

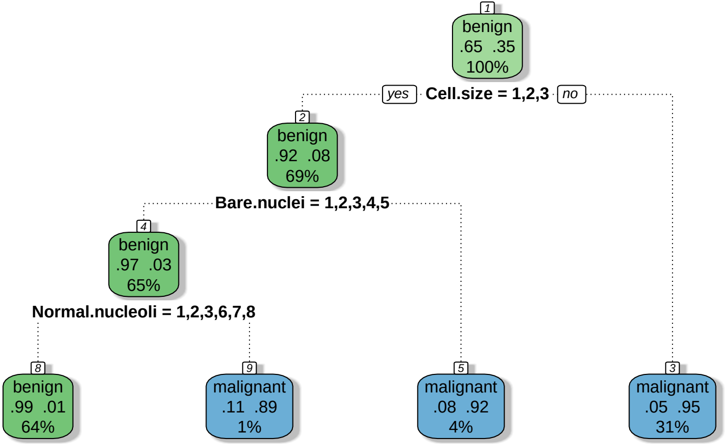
# DISADVANTAGES

Tree fitted on complete Wisconsin Breast Cancer data

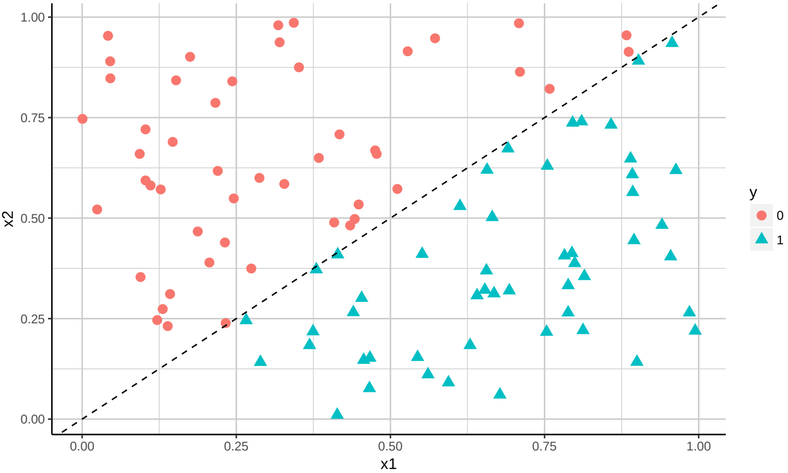


# DISADVANTAGES

Tree fitted on Wisconsin Breast Cancer data without observation 13

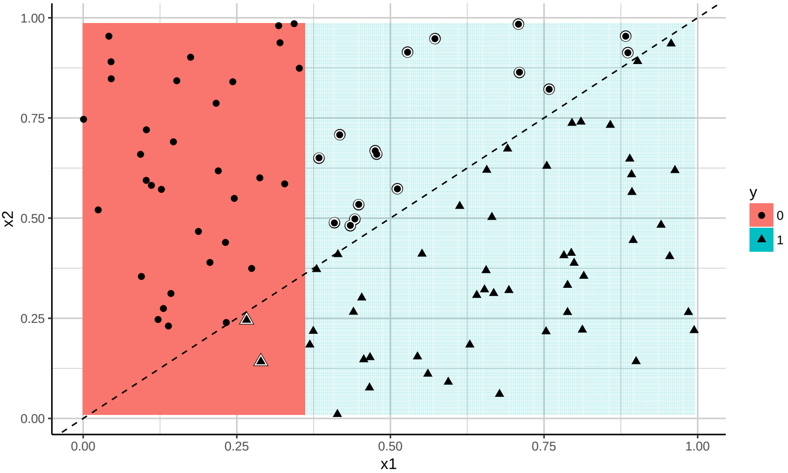


# DISADVANTAGES



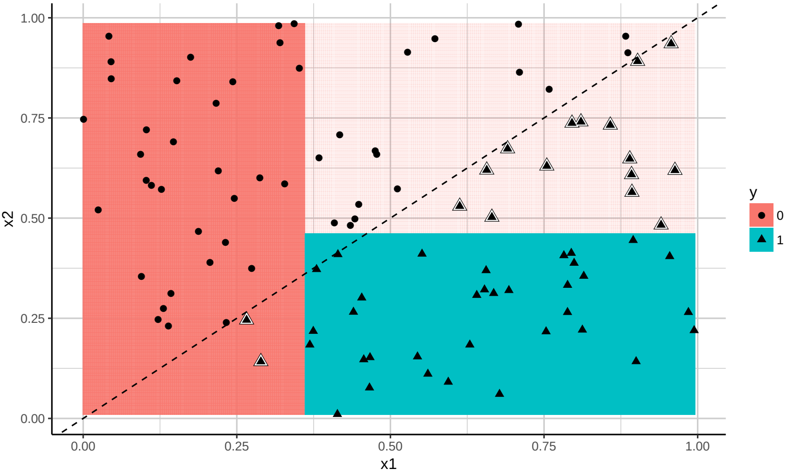
Linear dependencies must be modeled over several splits.

# DISADVANTAGES



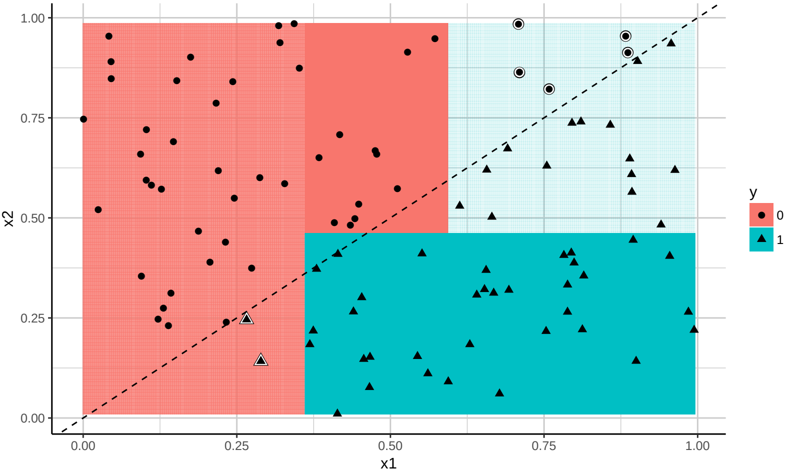
Linear dependencies must be modeled over several splits.

# DISADVANTAGES



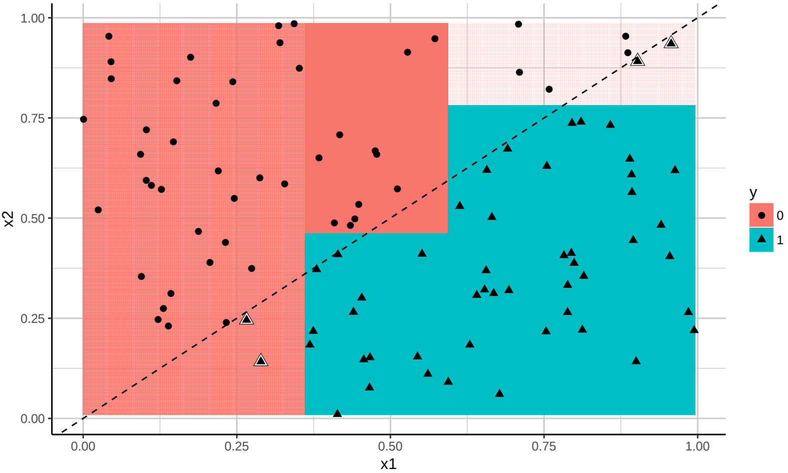
Linear dependencies must be modeled over several splits.

# DISADVANTAGES



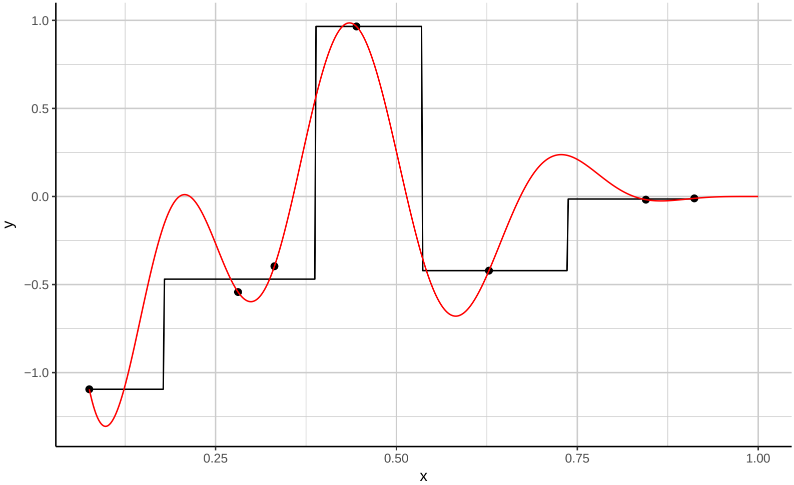
Linear dependencies must be modeled over several splits.

# DISADVANTAGES



Linear dependencies must be modeled over several splits.

# DISADVANTAGES



Prediction function isn’t smooth because a step function is fitted.