# **Introduction to Machine Learning**

CART Growing a Tree





#### Learning goals

- Understand how a tree is grown by an exhaustive search
- Know where and how the split point is set

- We start with an empty tree, a root node that contains all the data. Trees are then grown by recursively applying **greedy** optimization to each node  $\mathcal{N}$ .
- Greedy means we do an **exhaustive search**: Ideally, all possible splits of  $\mathcal{N}$  on all possible points *t* for all features  $x_j$  are compared in terms of their empirical risk  $\mathcal{R}(\mathcal{N}, j, t)$ .
- The training data is then distributed to child nodes according to the optimal split and the procedure is repeated in the child nodes.



× × 0 × × ×

- Start with a root node of all data.
- Search for feature and split point that minimizes the empirical risk in child nodes – makes label distribution more homogenous.



× 0 0 × 0 × ×

Proceed recursively for each child node: Select best split and divide data from parent node into left and right child nodes.





Repeat until we reach a stop criterion, e.g., until each leaf cannot be split further.



× 0 0 × 0 × ×

### SPLIT PLACEMENT



× × 0 × × ×

Splits are usually placed at the mid-point of the observations they split: the large margin to the next closest observations makes better generalization on new, unseen data more likely.

## FINDING THE SPLIT

Assume we split the data so that the misclassification error (MCE) is minimal through the splitting.

First, we check a set of potential splits for Sepal.Width



× 0 0 × 0 × ×

# FINDING THE SPLIT

Then we check a set of potential splits for Sepal.Length



× 0 0 × × ×

# FINDING THE SPLIT

- We take the split with lowest MCE: Sepal.Length = 5.5
- In real life, we actually search over many more splitting points. Common strategies involve: a) Searching over all possible split points (exhaustive search), b) searching quantile-wise
- MCE is rarely used, we will cover split criteria in detail later.



