Decision Trees

Dr. Daniele Cattaneo, J.-Prof. Dr. Josif Grabocka

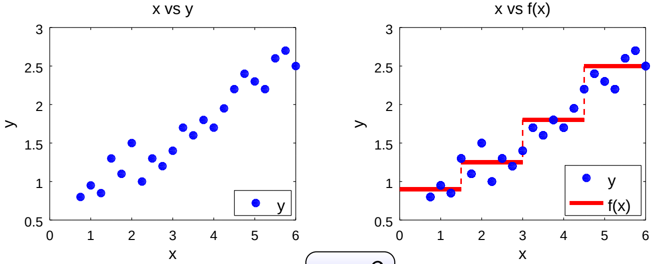
Machine Learning Course
Winter Semester 2023/2024

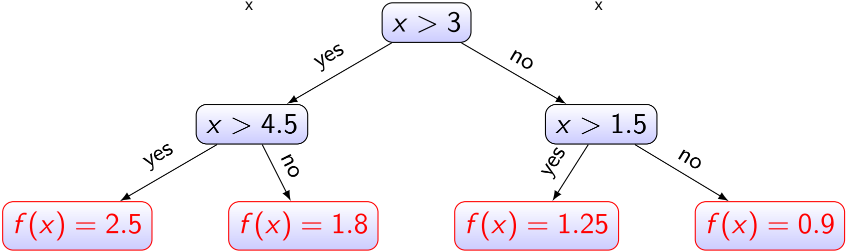
Albert-Ludwigs-Universität Freiburg

cattaneo@informatik.uni-freiburg.de, grabocka@informatik.uni-freiburg.de

November 13, 2023

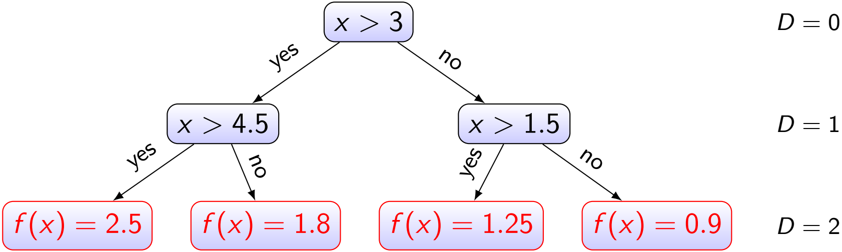
Step Functions as Prediction Models





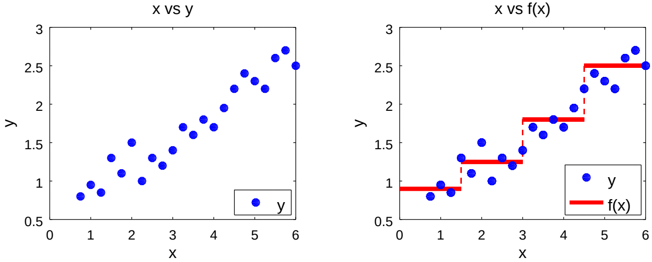
Legend: Decision Node Leaf

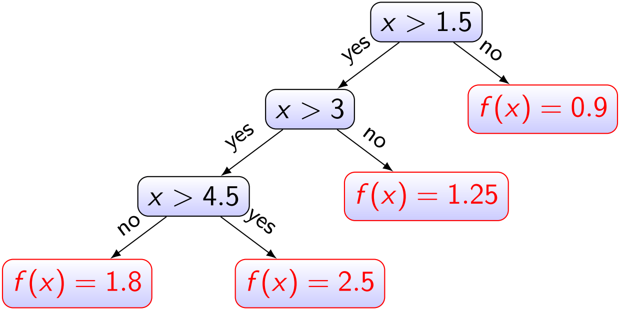
Trees — Depth and Height



The depth D of a node in a tree is the distance from the root node.
The height of a tree is the maximum depth of any node in the tree.

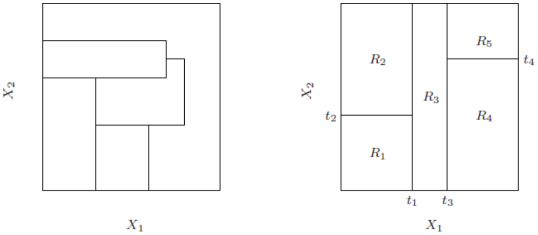
An Equivalent Tree





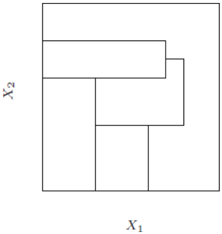
Feature space partitioning

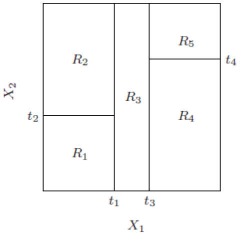
- Trees partition the feature space into regions



Feature space partitioning

- Trees partition the feature space into regions

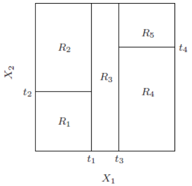




- In this lecture we will use binary splits
  - Every k-ary split can be seen as a sequence of binary splits
  - Binary splits are faster and often yield better predictions

Tree Predictions

- Consider a dataset $D := \{(x_i, y_i)\}_{i=1}^{N} \in (\mathcal{X} \times \mathcal{Y})^N$
- Partition the feature space $\mathcal{X}$ into regions $R_1, \ldots, R_J$



- For each R_j we aggregate a constant prediction model

$$\hat{y}^{(R_j)} = \text{aggregate}\left(\pi_y(\{(x, y) \in D \mid x \in R_j\})\right)$$

- Mathematically, a decision/regression tree $f(x)$ is specified by $\langle R_1, \ldots, R_J, \hat{y}^{(R_1)}, \ldots, \hat{y}^{(R_J)} \rangle$

Tree Prediction Model

Tree Prediction

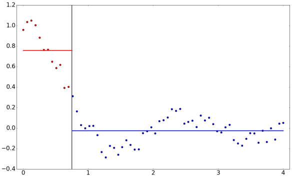
The prediction of a decision/regression tree with parameters $\theta = \langle R_1, \ldots, R_J, \hat{y}^{(R_1)}, \ldots, \hat{y}^{(R_J)} \rangle$ is

$$f(x, \theta) = \sum_{j=1}^{J} \hat{y}^{(R_j)} \, \mathbb{I}(x \in R_j)$$

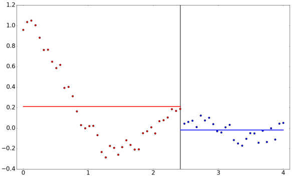
Here and throughout, $\mathbb{I}$ is the indicator function

$$\mathbb{I}(a) := \begin{cases} 1 & , \quad \text{if } a = \mathit{true} \\ 0 & , \quad \text{otherwise} \end{cases}$$

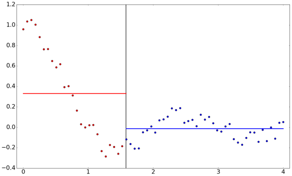
Intuition on splits: Regression



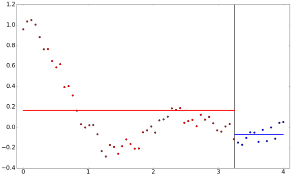
(black)



(red)

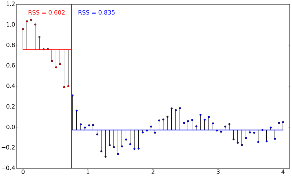


(yellow)

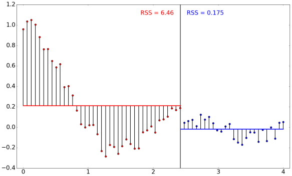


(blue)

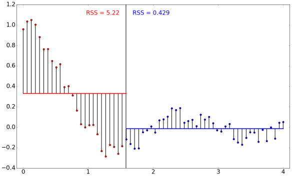
Intuition on splits: Regression



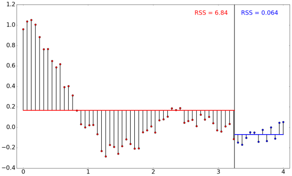
(black)



(red)



(yellow)



(blue)

Intuition on splits: Classification

Height (x) and Gender (y):

x	y
180	m
170	m
160	f
170	f
170	m
160	f
170	m

Split candidate: $x \leq 165 \rightsquigarrow$

$\{(160, f), (160, f)\} \in R_1$

$\{(180, m), (170, m), (170, f), (170, m), (170, m)\} \in R_2$

$\hat{y}^{(R_1)} := [P(y = m|x \leq 165) = 0, P(y = f|x \leq 165) = 1]$

$\hat{y}^{(R_2)} := [P(y = m|x > 165) = 0.8, P(y = f|x > 165) = 0.2]$

Split candidate: $x \leq 175 \rightsquigarrow$

$\{(170, m), (160, f), (170, f), (170, m), (160, f), (170, m)\} \in R_1$

$\{(180, m)\} \in R_2$

$\hat{y}^{(R_1)} := [P(y = m|x \leq 175) = 0.5, P(y = f|x \leq 175) = 0.5]$

$\hat{y}^{(R_2)} := [P(y = m|x > 175) = 1.0, P(y = f|x > 175) = 0.0]$

Answer: Splits that maximize purity

- A split $x \leq v$ divides $D := \{(x_i, y_i)\}_{i=1}^{N} \in (\mathcal{X} \times \mathcal{Y})^N$ into

  $D^{(\mathsf{L})} := \{(x, y) \in D \mid x \leq v\}$ and $D^{(\mathsf{R})} := \{(x, y) \in D \mid x > v\}$

- The impurity of a set is $H(D) : (\mathcal{X} \times \mathcal{Y})^{|D|} \to \mathbb{R}_+$

- The best split $x \leq v$ maximizes the gain $I(x \leq v)$ in purity:

$$I(x \leq v) = \underbrace{|D| \cdot H(D)}_{\text{Impurity before split}} - \underbrace{\left(|D^{(\mathsf{L})}| \cdot H\left(D^{(\mathsf{L})}\right) + |D^{(\mathsf{R})}| \cdot H\left(D^{(\mathsf{R})}\right)\right)}_{\text{Impurity after split}}$$

Exhaustive Search of Decision Splits

- Mid-points of the unique values of the j-th feature are $V^{(j)}$
  - E.g. In a dataset $D \in \left(\mathbb{N}^2 \times \{0,1\}\right)^N$ with values $D := \{((1,5),1),((3,4),0),((1,6),1),((5,5),1),((4,8),1)\}$
  - Then $V^{(1)} = \{2, 3.5, 4.5\}$, $V^{(2)} = \{4.5, 5.5, 7\}$
  - Categorical features can be converted to numerical ones

- The optimal split $x_{:,j} \leq v$ in a dataset with M features is:

$$\underset{\substack{j \in \{1,\ldots,M\} \\ v \in V^{(j)}}}{\operatorname{argmax}} \; I(x_{:,j} \leq v)$$

- Runtime complexity is $\mathcal{O}\left(N^2 M\right)$ when implemented naively

Classification and Regression Trees

Algorithm 1: CART - Train Decision Tree

Data: *Params*: Set $D := \{(x_i, y_i)\}_{i=1}^{|D|}$, Node n, Depth ℓ
Hyper-params: Max Depth: L, Min Instances for Split: δ
if $|D| > \delta \wedge \ell \leq L$ **then**
    $(x_{:,j}, v) :=$ SearchSplit(D) ;   `// Find best split`
    Decision node: $n \leftarrow (x_{:,j} \leq v,\ yes/no?)$;
    $D^{(\mathrm{L})} := \{(x, y) \in D \mid x_{:,j} \leq v\}$;
    CART($D^{(\mathrm{L})}$, n.left, $\ell + 1$) ;   `// Create left sub-tree`
    $D^{(\mathrm{R})} := \{(x, y) \in D \mid x_{:,j} > v\}$;
    CART($D^{(\mathrm{R})}$, n.right, $\ell + 1$);   `// Create right sub-tree`
else
    Leaf $n \leftarrow \hat{y} = \text{aggregate}\,(\pi_y(D))$;
end

Call initially: $\text{CART}(D, \text{Root Node}, 1)$

Regression: Impurity = Variance

For a split $x \leq v$ dividing D into $D^{(\mathsf{L})}, D^{(\mathsf{R})}$

The predicted target is the mean target of the split subset:

$$\hat{y}^{(\mathsf{L})}(x) = \frac{1}{|D^{(\mathsf{L})}|} \sum_{(x,y) \in D^{(\mathsf{L})}} y \quad \text{and} \quad \hat{y}^{(\mathsf{R})}(x) = \frac{1}{|D^{(\mathsf{R})}|} \sum_{(x,y) \in D^{(\mathsf{R})}} y$$

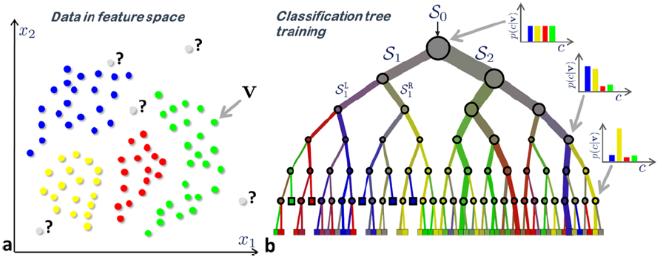
The impurity of the split subsets are the variances:

$$|D^{(\mathsf{L})}| \cdot H(D^{(\mathsf{L})}) = \sum_{(x,y) \in D^{(\mathsf{L})}} \left(y - \hat{y}^{(\mathsf{L})}(x) \right)^2$$

$$|D^{(\mathsf{R})}| \cdot H(D^{(\mathsf{R})}) = \sum_{(x,y) \in D^{(\mathsf{R})}} \left(y - \hat{y}^{(\mathsf{R})}(x) \right)^2$$

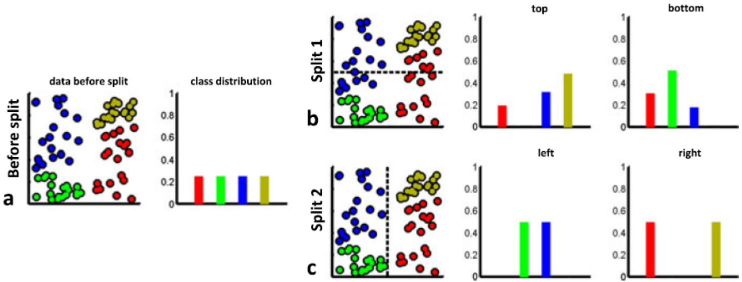
Best split for regression: $\underset{x \leq v}{\text{argmin}} \; |D^{(\mathsf{L})}| \cdot H(D^{(\mathsf{L})}) + |D^{(\mathsf{R})}| \cdot H(D^{(\mathsf{R})})$

Classification Trees



- Conceptually the same as for regression
- Leaf model: majority vote; probability of data in the leaf
- Split criterion: Gini index; variance reduction; information gain

Intuition on Classification Purity



Classification: Impurity=Entropy

Definition: Entropy

The entropy of a discrete random variable V with K possible outcomes v_k of respective probability $p(v_k)$ (for $k = 1, \ldots, K$) is

$$H(V) = -\sum_{k=1}^{K} p(v_k) \log_2 p(v_k)$$

Classification: Impurity=Entropy

- Distance to the worst-case $q(v_k) = \frac{1}{K}, \forall k \in \{1, \ldots, K\}$ is:

$$\begin{aligned} KL(p, q) &= \sum_{k=1}^{K} p(v_k) \log_2 \left(\frac{p(v_k)}{q(v_k)} \right) \\ &= \sum_{k=1}^{K} p(v_k) \log_2 \left(\frac{p(v_k)}{\frac{1}{K}} \right) \\ &= \sum_{k=1}^{K} p(v_k) \log_2 (p(v_k)) + \log_2 (K) \sum_{k=1}^{K} p(v_k) \end{aligned}$$

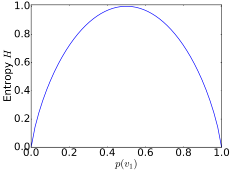
- Therefore: $\max KL(p, q) = \min(-KL(p, q)) = \min H(V)$

Classification: Impurity=Entropy

Definition: Entropy

The entropy of a discrete random variable V with K possible outcomes v_k of respective probability $p(v_k)$ (for $k = 1, \ldots, K$) is

$$H(V) = -\sum_{k=1}^{K} p(v_k) \log_2 p(v_k)$$



Examples for a Boolean random variable V with outcomes v_1 and v_2:

- $p(v_1) = 0.5$, $p(v_2) = 0.5$. Then,
  $H(V) = -0.5 \log_2(0.5) - 0.5 \log_2(0.5) = 1$
- $p(v_1) = 0$, $p(v_2) = 1$. Then,
  $H(V) = -\lim_{x \to 0} x \log_2(x) - 1 \log_2(1) = 0$

Information Gain

- Split $x \leq v$ divide D into $D^{(L)}$ and $D^{(R)}$
- $p(V)$ is the distribution of the N target values of D
- $p\left(V^{(L)}\right)$ is the distribution of the $N^{(L)}$ target values of $D^{(L)}$
- $p\left(V^{(R)}\right)$ is the distribution of the $N^{(R)}$ target values of $D^{(R)}$
- The Information Gain of the split $x \leq v$ is:

$$
\begin{aligned}
I(x \leq v) =& N \cdot H(V) - N^{(L)} \cdot H\left(V^{(L)}\right) - N^{(R)} \cdot H\left(V^{(R)}\right) \\
=& -N \sum_{v_k \in V} p\left(v_k\right) \log_2 p\left(v_k\right) \\
& + N^{(L)} \sum_{v_k^{(L)} \in V^{(L)}} p\left(v_k^{(L)}\right) \log_2 p\left(v_k^{(L)}\right) \\
& + N^{(R)} \sum_{v_k^{(R)} \in V^{(R)}} p\left(v_k^{(R)}\right) \log_2 p\left(v_k^{(R)}\right)
\end{aligned}
$$

Information Gain: Example

$$I(x \leq v) = N \cdot H(V) - N^{(L)} \cdot H\left(V^{(L)}\right) - N^{(R)} \cdot H\left(V^{(R)}\right)$$

- Example: Splitting 8 emails ($4 \times$ spam, $4 \times$ non-spam).
  - $V = \{\text{spam}, \text{non-spam}\}, p(V) = \{0.5, 0.5\}, N \cdot H(V) = 8 \cdot 1 = 8$
  - Split 1: Left: $4 \times \text{spam}, 0 \times \text{non-spam}$; Right: $0 \times \text{spam}, 4 \times \text{non-spam}$
    - $p(V^{(L)}) = \{1.0, 0.0\}, p(V^{(R)}) = \{0.0, 1.0\}$
    - $N^{(L)} \cdot H\left(V^{(L)}\right) = 4 \cdot 0 = 0, \;\; N^{(R)} \cdot H\left(V^{(R)}\right) = 4 \cdot 0 = 0$
    - $I = 8 - 0 - 0 = 8$ (maximal gain in purity)
  - Split 2: Left: $2 \times$ spam, $2 \times$ non-spam; Right: $2 \times$ spam, $2 \times$ non-spam
    - $p(V^{(L)}) = \{0.5, 0.5\}, p(V^{(R)}) = \{0.5, 0.5\}$
    - $N^{(L)} \cdot H\left(V^{(L)}\right) = 4 \cdot 1 = 4, \;\; N^{(R)} \cdot H\left(V^{(R)}\right) = 4 \cdot 1 = 4$
    - $I = 8 - 4 - 4 = 0$ (no gain in purity)

A Classification Example: Revisited (I)

$$I(x \leq v) = N \cdot H(V) - N^{(L)} \cdot H\left(V^{(L)}\right) - N^{(R)} \cdot H\left(V^{(R)}\right)$$

Height (x) and Gender (y):

x	y
180	m
170	m
160	f
170	f
170	m
160	f
170	m

- $K = 2, N = 7, V = \{m, f\}, p(V) = \{0.57, 0.43\}$

  $H(V) = -(0.57 \log_2 (0.57) + 0.43 \log_2 (0.43)) \approx 0.98$

- Split candidate $x \leq 165$

  $\{(160, f), (160, f)\} \in R^{(L)}$

  $\{(180, m), (170, m), (170, f), (170, m), (170, m)\} \in R^{(R)}$

  $p(V^{(L)}) = \{1.0, 0.0\}$ and $p(V^{(R)}) = \{0.8, 0.2\}$

  $H\left(V^{(L)}\right) = -(1 \log_2 (1) + 0 \log_2 (0)) = 0$

  $H\left(V^{(R)}\right) = -(0.8 \log_2 (0.8) + 0.2 \log_2 (0.2)) \approx 0.72$

  $I(x \leq 165) = 7 \cdot 0.98 - 2 \cdot 0 - 5 \cdot 0.72 = 3.92$

A Classification Example: Revisited (II)

$$I(x \leq v) = N \cdot H(V) - N^{(L)} \cdot H\left(V^{(L)}\right) - N^{(R)} \cdot H\left(V^{(R)}\right)$$

Height (x) and Gender (y):

x	y
180	m
170	m
160	f
170	f
170	m
160	f
170	m

- $K = 2, N = 7, V = \{m, f\}, p(V) = \{0.57, 0.43\}$
- Split candidate $x \leq 175$

$\{(170, m), (160, f), (170, f), (170, m), (160, f), (170, m)\} \in R^{(L)}$

$\{(180, m)\} \in R^{(R)}$

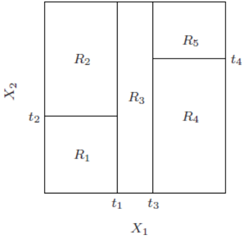
$p(V^{(L)}) = \{0.5, 0.5\}$ and $p(V^{(R)}) = \{1.0, 0.0\}$

$H\left(V^{(L)}\right) = -\left(0.5 \log_2(0.5) + 0.5 \log_2(0.5)\right) = 1.0$

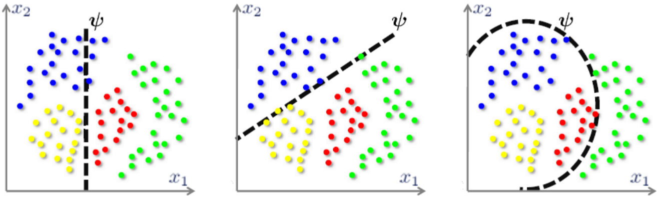
$H\left(V^{(R)}\right) = -\left(1.0 \log_2(1.0) + 0.0 \log_2(0.0)\right) = 0.0$

$I(x \leq 175) = 7 \cdot 0.98 - 6 \cdot 1 - 1 \cdot 0.0 = 0.9$

Alternative Types of Splits



Axis-aligned split are standard, but more complex splits are possible



Pros and Cons of Decision and Regression Trees

- \+ Flexible framework with exchangeable components: splitting criterion, leaf model, type of split
- \+ Interpretability
- \+ Handle categorical input values natively
- \+ Handle unimportant features well
- \+ Scalable for large datasets
- \- Tend to overfit
- \- Deterministic, i.e. not suitable for some ensemble methods

Hyperparameters of Decision and Regression Trees

Regression and decision trees have several hyperparameters:

- Minimum number of samples in a leaf (`min_leaf`)
- Maximal depth of the tree (`max_depth`)
- Total number of nodes
- Leaf model (weak learner; here constant)
- Split criterion

Bias and Variance of Trees

- Facts about tree-based models
  - Trees are very expressive models
  - When you slightly change the data, you might get a very different tree
- As a result:
  - Trees are a high-variance model.
  - Trees are a low-bias model.

Computational Complexity of CART

- N data points of dimensionality M, axis-aligned splits
- Finding the best split value for a given feature (pre-sorted): $O(N)$
  - Amortized analysis: $O(N)$ to initialize all data points to the left child, $O(1)$ for moving one data point from left to right at a time
- Finding best split point at root: $O(MN)$
- Let $T(N)$ denote the time we pay for a complete subtree with N data points; this has a part due to the cost at the root and parts due to the 2 smaller child subtrees
- Best case: balanced trees
  - Fitting: $T(N) = O(MN) + 2T(N/2)$
  - $\rightsquigarrow$ This leads to $O(MN \log N)$ since we pay $O(M \cdot N)$ at each of $\log N$ levels
  - Prediction: $O(\log N)$

Computational Complexity of CART

- N data points of dimensionality M, axis-aligned splits
- Finding the best split value for a given feature (pre-sorted): $O(N)$
  - Amortized analysis: $O(N)$ to initialize all data points left to the left child, $O(1)$ for moving one data point from left to right at a time
- Finding best split point at root: $O(MN)$
- Let $T(N)$ denote the time we pay for a complete subtree with N data points; this has a part due to the cost at the root and parts due to the 2 smaller child subtrees
- Worst case: splitting off one data point at a time
  - Fitting: $T(N) = O(MN) + T(N-1)$; this leads to $O(MN^2)$
  - Prediction: $O(N)$