Supervised learning

- Cluster analysis and association rules are not concerned with a specific target attribute.
- Supervised learning refers to problems where the value of a target attribute should be predicted based on the values of other attributes.
- Problems with a categorical target attribute are called classification, problems with a numerical target attribute are called regression.

Finding explanations

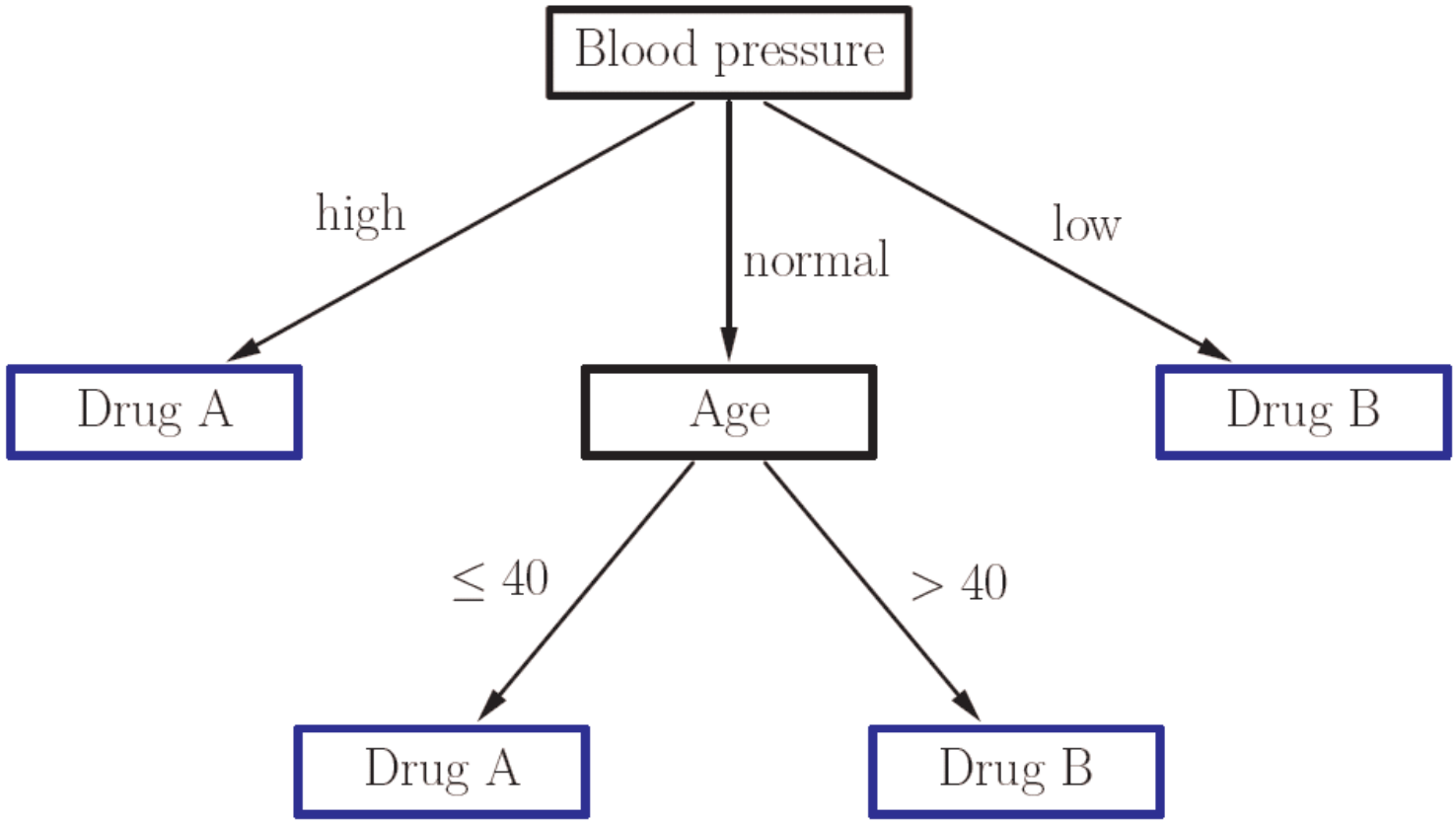
- Attributes: Class C, other attributes $A^{(1)}, \ldots, A^{(m)}$
- Data: $\mathcal{S} = \{(\mathbf{x}_i, c_i) | i = 1, \ldots, N\}$
- Finding interpretable model to understand dependency of target attribute c_i and the input vectors $\mathbf{x}_i$.
- Model will not express necessarily the causal relationship, but only numerical correlations.

Decision trees

- Find hierarchical structure to explain how different areas in the input space correspond to different outcomes
- Useful for data with a lot of attributes of unknown importance
- Insensitive to normalization issues
- Tolerant to correlated and noisy attributes

A very simple decision tree

Assignment of a drug to a patient:



Classification with decision trees

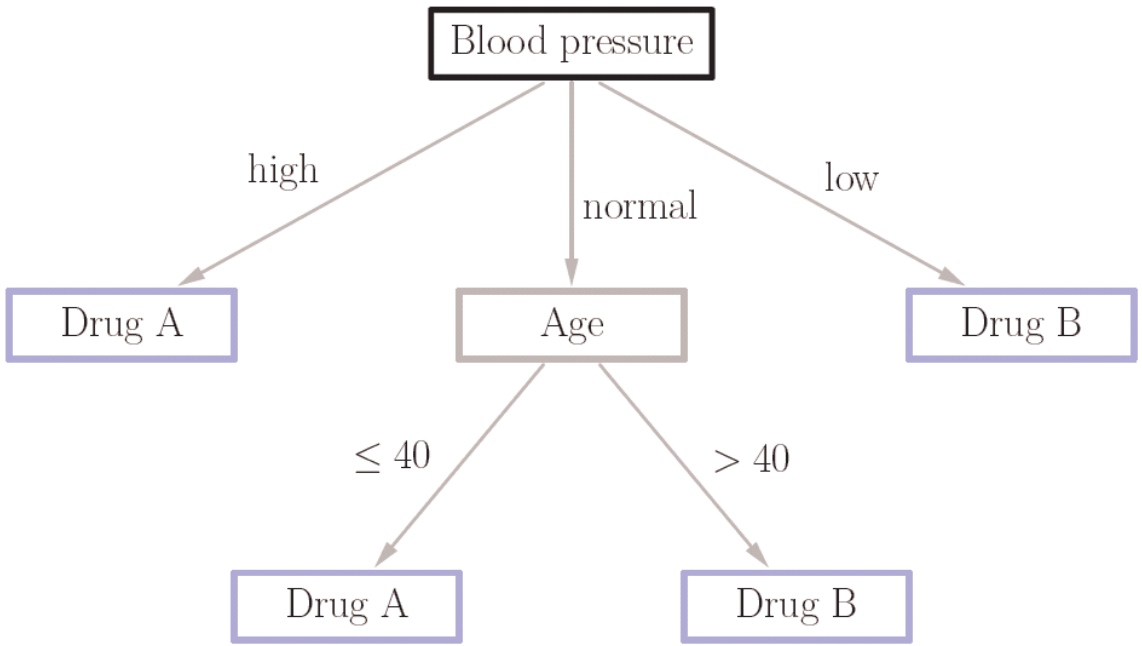
Recursive Descent:

- Start at the root node.
- If the current node is an **leaf node**:
  - Return the class assigned to the node.
- If the current node is an **inner node**:
  - Test the attribute associated with the node.
  - Follow the branch labeled with the outcome of the test.
  - Apply the algorithm recursively.

Intuitively: Follow the path corresponding to the case to be classified.

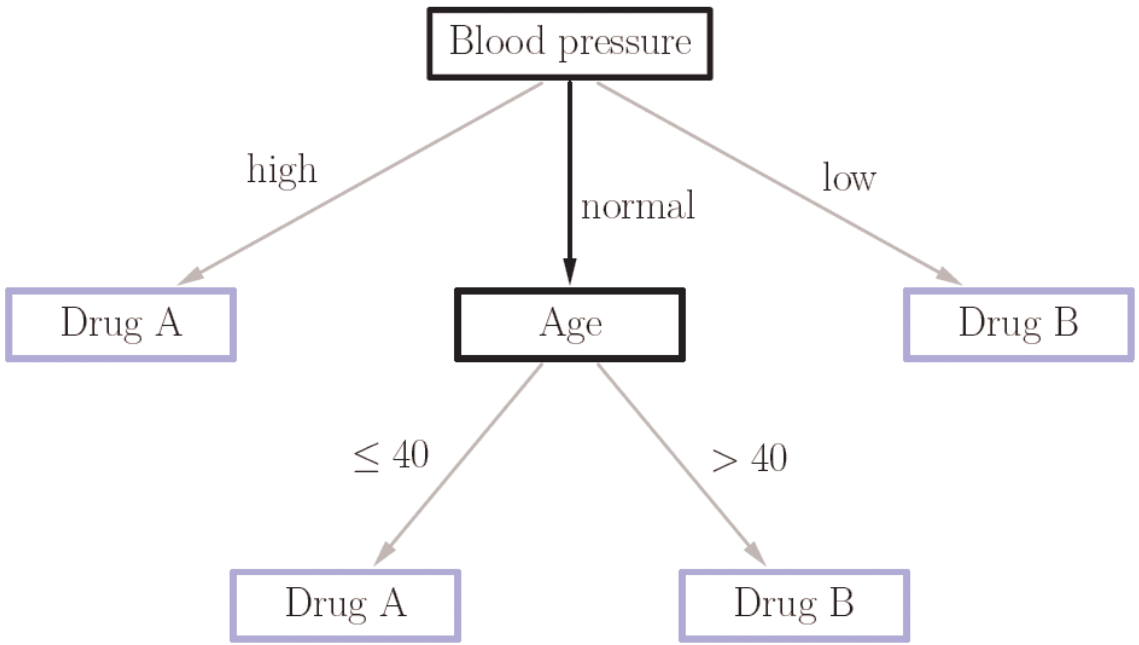
Classification with decision trees

Assignment of a drug to a 30 year old patient with normal blood pressure:



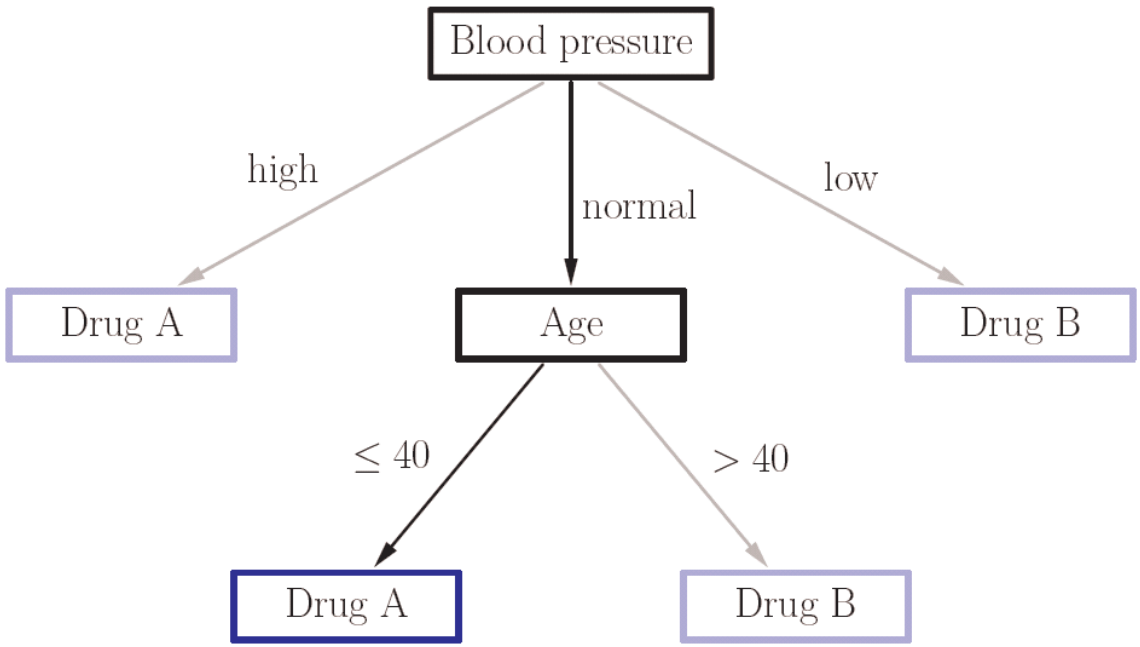
Classification with decision trees

Assignment of a drug to a 30 year old patient with normal blood pressure:



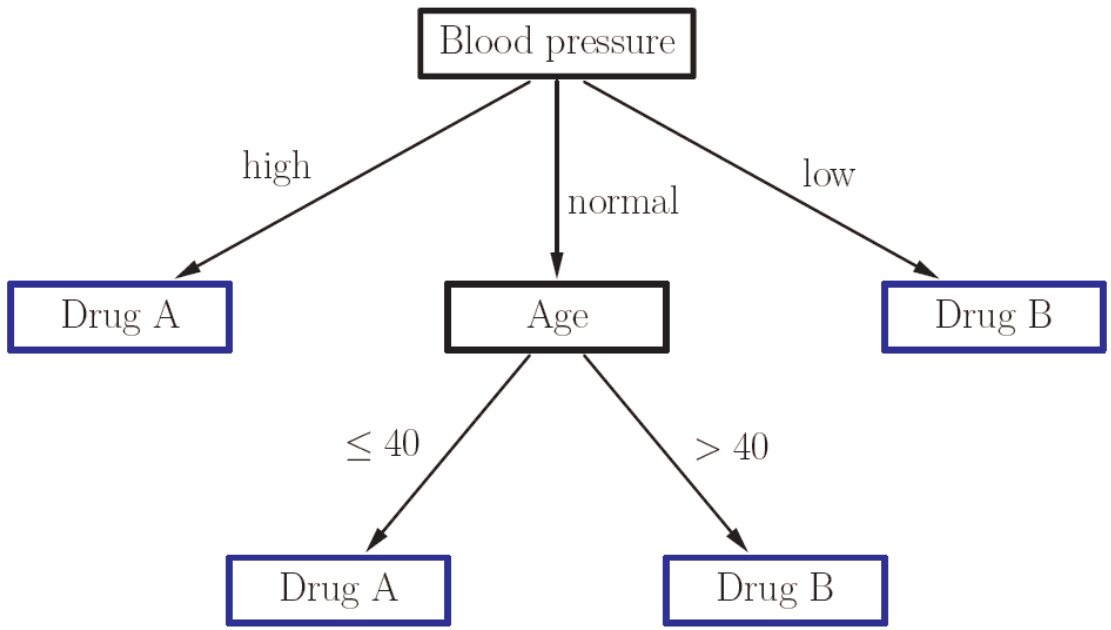
Classification with decision trees

Assignment of a drug to a 30 year old patient with normal blood pressure:



Disjunction of conjunctions

- Drug A $\Leftrightarrow$ Blood pressure $=$ high
  $\vee$ Blood pressure $=$ normal $\wedge$ Age ≤ 40
- Drug B $\Leftrightarrow$ Blood pressure $=$ low
  $\vee$ Blood pressure $=$ normal $\wedge$ Age > 40



Induction of decision trees

- **Top-down approach**
  - Build the decision tree from top to bottom (from the root to the leaves).
- **Greedy selection of a test attribute**
  - Compute an evaluation measure for all attributes.
  - Select the attribute with the best evaluation.
- **Divide and conquer / recursive descent**
  - Divide the example cases according to the values of the test attribute.
  - Apply the procedure recursively to the subsets.
  - Terminate the recursion if
    - all cases belong to the same class or
    - no more test attributes are available

Decision tree induction: Example

Patient database

- 12 example cases
- 3 descriptive attributes
- 1 class attribute

Assignment of drug

(without patient attributes)

always drug A or always drug B:

50% correct (in 6 of 12 cases)

No	Sex	Age	Blood pr.	Drug
1	male	20	normal	A
2	female	73	normal	B
3	female	37	high	A
4	male	33	low	B
5	female	48	high	A
6	male	29	normal	A
7	female	52	normal	B
8	male	42	low	B
9	male	61	normal	B
10	female	30	normal	A
11	female	26	low	B
12	male	54	high	A

Decision tree induction: Example

Sex of the patient

- Division w.r.t. male/female.

Assignment of drug

male:	50% correct	(in 3 of 6 cases)
female:	50% correct	(in 3 of 6 cases)
total:	**50% correct**	(in 6 of 12 cases)

No	Sex	Drug
1	male	A
6	male	A
12	male	A
4	male	B
8	male	B
9	male	B
3	female	A
5	female	A
10	female	A
2	female	B
7	female	B
11	female	B

Decision tree induction: Example

Blood pressure of the patient

- Division w.r.t. high/normal/low.

Assignment of drug

high:	A	100% correct	(in 3 of 3 cases)
normal:		50% correct	(in 3 of 6 cases)
low:	B	100% correct	(in 3 of 3 cases)
total:		**75% correct**	(in 9 of 12 cases)

No	Blood pr.	Drug
3	high	A
5	high	A
12	high	A
1	normal	A
6	normal	A
10	normal	A
2	normal	B
7	normal	B
9	normal	B
4	low	B
8	low	B
11	low	B

Decision tree induction: Example

Age of the patient

- Sort according to age.
- Find best age split.
  here: ca. 40 years

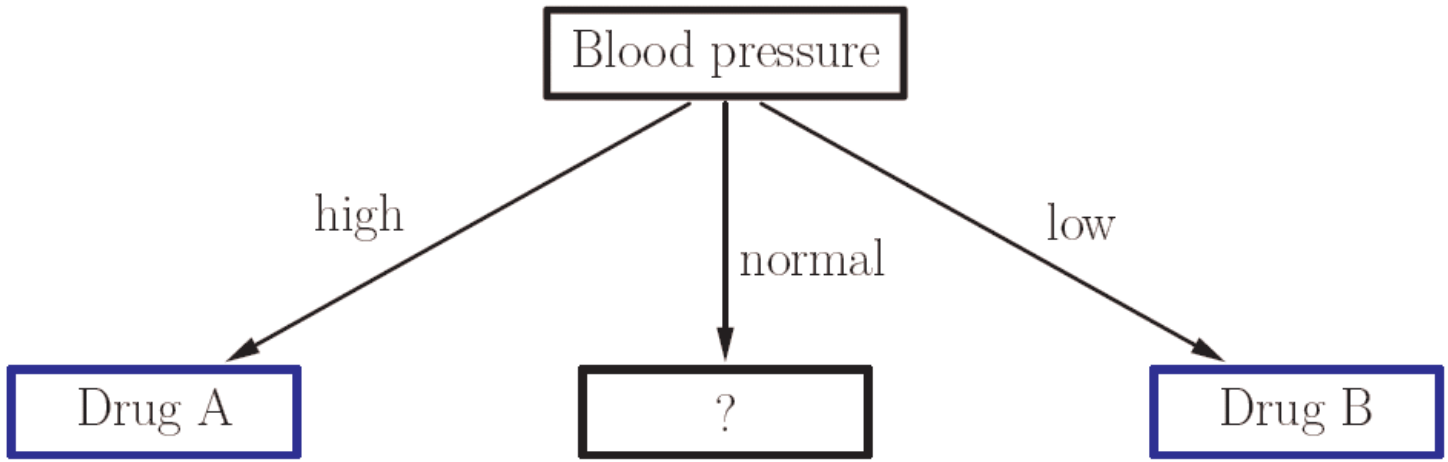
Assignment of drug

≤ 40:	A	67% correct	(in 4 of 6 cases)
> 40:	B	67% correct	(in 4 of 6 cases)
total:		**67% correct**	(in 8 of 12 cases)

No	Age	Drug
1	20	A
11	26	B
6	29	A
10	30	A
4	33	B
3	37	A
8	42	B
5	48	A
7	52	B
12	54	A
9	61	B
2	73	B

Decision tree induction: Example

Current decision tree:



Decision tree induction: Example

Blood pressure and sex

- Only patients with normal blood pressure.
- Division w.r.t. male/female.

No	Blood pr.	Sex	Drug
3	high		A
5	high		A
12	high		A
1	normal	male	A
6	normal	male	A
9	normal	male	B
2	normal	female	B
7	normal	female	B
10	normal	female	A
4	low		B
8	low		B
11	low		B

Assignment of drug

male:	A	67% correct	(2 of 3)
female:	B	67% correct	(2 of 3)
total:		**67% correct**	(4 of 6)

Decision tree induction: Example

Blood pressure and age

- Only patients with normal blood pressure.
- Sort according to age.
- Find best age split. here: ca. 40 years

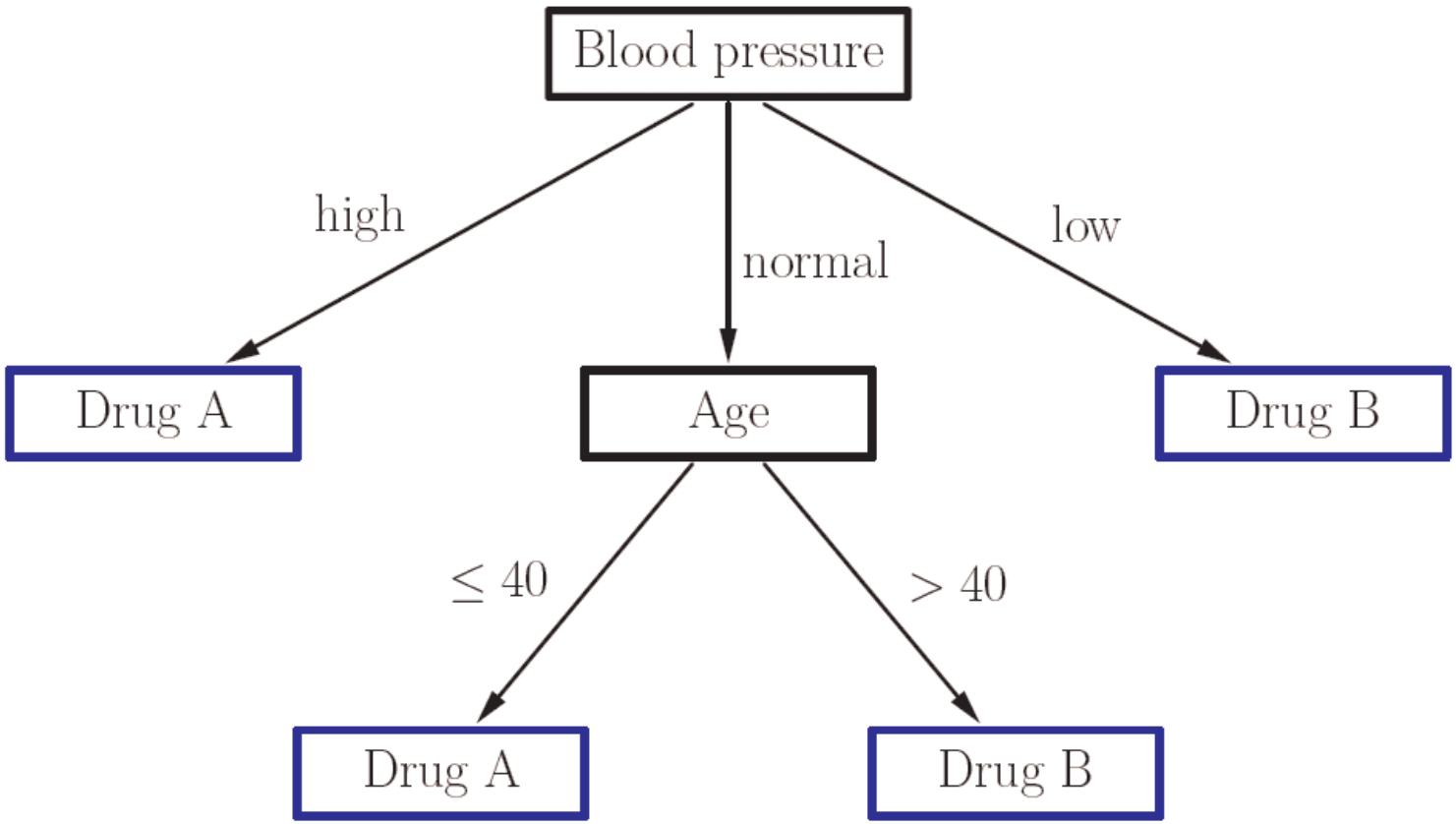
No	Blood pr.	Age	Drug
3	high		A
5	high		A
12	high		A
1	normal	20	A
6	normal	29	A
10	normal	30	A
7	normal	52	B
9	normal	61	B
2	normal	73	B
11	low		B
4	low		B
8	low		B

Assignment of drug

≤ 40:	A	100% correct	(3 of 3)
> 40:	B	100% correct	(3 of 3)
total:		**100% correct**	(6 of 6)

Decision tree induction: Example

Resulting decision tree:



Decision tree induction: Notation

S	a set of case or object descriptions
C	the class attribute
$A^{(1)}, \ldots, A^{(m)}$	other attributes (index dropped in the following)
$\mathrm{dom}(C)$	$= \{c_1, \ldots, c_{n_C}\}$, n_C: number of classes
$\mathrm{dom}(A)$	$= \{a_1, \ldots, a_{n_A}\}$, n_A: number of attribute values
$N_{..}$	total number of case or object descriptions i.e. $N_{..} = \lvert S \rvert$
$N_{i.}$	absolute frequency of the class c_i
$N_{.j}$	absolute frequency of the attribute value a_j
N_{ij}	absolute frequency of the combination of the class c_i and the attribute value a_j. $N_{i.} = \sum_{j=1}^{n_A} N_{ij}$ and $N_{.j} = \sum_{i=1}^{n_C} N_{ij}$.
$p_{i.}$	relative frequency of the class c_i, $p_{i.} = \frac{N_{i.}}{N_{..}}$
$p_{.j}$	relative frequency of the attribute value a_j, $p_{.j} = \frac{N_{.j}}{N_{..}}$
p_{ij}	relative frequency of the combination of class c_i and attribute value a_j, $p_{ij} = \frac{N_{ij}}{N_{..}}$
$p_{i\lvert j}$	relative frequency of the class c_i in cases having attribute value a_j, $p_{i\lvert j} = \frac{N_{ij}}{N_{.j}} = \frac{p_{ij}}{p_{.j}}$

Principle of decision tree induction

function grow_tree (S : set of cases) : node;
begin
    best_v := WORTHLESS;
    for all untested attributes A **do**
        compute frequencies N_{ij}, $N_{i.}$, $N_{.j}$ for $1 \leq i \leq n_C$ and $1 \leq j \leq n_A$;
        compute value v of an evaluation measure using N_{ij}, $N_{i.}$, $N_{.j}$;
        if $v > best\_v$ **then** *best_v* := v; *best_A* := A; **end**;
    end
    if $best\_v =$ WORTHLESS
    then create leaf node x;
        assign majority class of S to x;
    else create test node x;
        assign test on attribute *best_A* to x;
        for all $a \in \mathrm{dom}(best\_A)$ **do** x.child[a] := grow_tree($S|_{best\_A=a}$); **end**;
    end;
    return x;
end;

Evaluation Measures

- Evaluation measure used in the above example:
  rate of correctly classified example cases.
  - Advantage: simple to compute, easy to understand.
  - Disadvantage: works well only for two classes.
- If there are more than two classes, the rate of misclassified example cases **neglects a significant amount of the available information**.
  - Only the majority class—that is, the class occurring most often in (a subset of) the example cases—is really considered.
  - The distribution of the other classes has no influence. However, a good choice here can be important for deeper levels of the decision tree.

Evaluation measures

- **Therefore:** Study also other evaluation measures. Here:
  - **Information gain** and its various normalisations.
  - **Gini index**
  - χ^2 **measure**

An Information-theoretic Evaluation Measure

Information Gain (Kullback/Leibler 1951, Quinlan 1986)

Based on Shannon Entropy $H = -\sum_{i=1}^{n} p_i \log_2 p_i$

$$
\begin{aligned}
I_{\text{gain}}(C, A) &= \overbrace{H(C)}^{} - \overbrace{H(C|A)}^{} \\
&= \overbrace{-\sum_{i=1}^{n_C} p_{i.} \log_2 p_{i.}}^{} - \overbrace{\sum_{j=1}^{n_A} p_{.j} \left(-\sum_{i=1}^{n_C} p_{i|j} \log_2 p_{i|j} \right)}^{}
\end{aligned}
$$

$H(C)$ Entropy of the class distribution (C: class attribute)

$H(C|A)$ Expected entropy of the class distribution if the value of the attribute A becomes known

$H(C) - H(C|A)$ Expected entropy reduction or information gain

Interpretation of Shannon entropy

- Let $S = \{s_1, \ldots, s_n\}$ be a finite set of alternatives having positive probabilities $P(s_i)$, $i = 1, \ldots, n$, satisfying $\sum_{i=1}^{n} P(s_i) = 1$.

- **Shannon Entropy:**

$$H(S) = -\sum_{i=1}^{n} P(s_i) \log_2 P(s_i)$$

- Intuitively: **Expected number of yes/no questions that have to be asked in order to determine the obtaining alternative.**

Entropy

Entropy can be interpreted as a measure for the information gained by knowing the outcome of a random experiment.

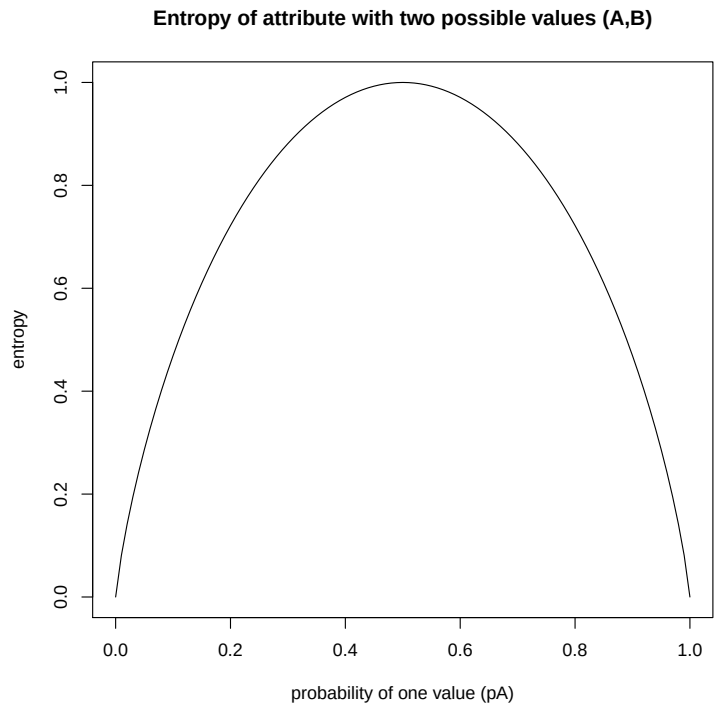
Example. When a (fair) coin is thrown, then chance to guess the outcome correctly is 50%.
The experiment has only two outcomes (head (0) or tail (1)).

The information gained by knowing the outcome is half a bit, since one would have 50% of the cases correctly by pure guessing.

Entropy

For a very asymmetric distribution (an unfair coin), the information gained by knowing the outcome (entropy) is much smaller. By guessing the more probable outcome, one will have more than 50% correct guesses.

In the extreme case of a certain event (e.g. a coin that is manipulated in such a way that it will always show tail), the outcome can always be predicted correctly and the entropy is 0.



Interpretation of information gain

- **Information Gain**:

$$I_{\text{gain}}(C, A) = H(C) - H(C|A)$$

- Intuitively: **Impurity that is left in the subsets of the original data after they have been split according to their values of A.**

Example

ID	**Height**	Weight	Long hair	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

Example

ID	**Height**	Weight	Long hair	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

$$H(\text{Sex}) = -\left(\tfrac{4}{10}\log_2\left(\tfrac{4}{10}\right) + \tfrac{6}{10}\log_2\left(\tfrac{6}{10}\right)\right) \approx 0.9710$$

Example

ID	**Height**	Weight	Long hair	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

$$H(\text{Sex—Height}) = -\tfrac{4}{10}\left(\tfrac{3}{4}\log_2\left(\tfrac{3}{4}\right) + \tfrac{1}{4}\log_2\left(\tfrac{1}{4}\right)\right) - \ldots$$

Example

ID	**Height**	Weight	Long hair	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

$$H(\text{Sex—Height}) = \ldots - \tfrac{3}{10}\left(\tfrac{2}{3}\log_2\left(\tfrac{2}{3}\right) + \tfrac{1}{3}\log_2\left(\tfrac{1}{3}\right)\right) - \ldots$$

Example

ID	**Height**	Weight	Long hair	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

$$H(\text{Sex—Height}) = \ldots - \tfrac{3}{10}\left(\tfrac{1}{3}\log_2\left(\tfrac{1}{3}\right) + \tfrac{2}{3}\log_2\left(\tfrac{2}{3}\right)\right)$$

Example

$$
\begin{aligned}
H(\text{Sex—Height}) &= -\frac{4}{10}\left(\frac{3}{4}\log_2\left(\frac{3}{4}\right)+\frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) \\
&\quad -\frac{3}{10}\left(\frac{2}{3}\log_2\left(\frac{2}{3}\right)+\frac{1}{3}\log_2\left(\frac{1}{3}\right)\right) \\
&\quad -\frac{3}{10}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right)+\frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) \\
&\approx 0.8755
\end{aligned}
$$

Example

ID	Height	**Weight**	Long hair	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

$$H\big(\text{Sex—Weight}\big) = -\tfrac{3}{10} \cdot 0 - \ldots$$

Example

ID	Height	**Weight**	Long hair	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

$$H(\text{Sex—Weight}) = \ldots - \tfrac{5}{10}\left(\tfrac{3}{5}\log_2\left(\tfrac{3}{5}\right) + \tfrac{2}{5}\log_2\left(\tfrac{2}{5}\right)\right) - \ldots$$

Example

ID	Height	**Weight**	Long hair	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

$$H\big(\text{Sex—Weight}\big) = \ldots - \tfrac{2}{10} \cdot 0$$

Example

$$
\begin{aligned}
H(\text{Sex—Weight}) &= -\frac{3}{10} \cdot 0 \\
&\quad -\frac{5}{10}\left(\frac{3}{5}\log_2\left(\frac{3}{5}\right) + \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right) \\
&\quad -\frac{2}{10} \cdot 0 \\
&\approx 0.4855
\end{aligned}
$$

Example

ID	Height	Weight	**Long hair**	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

$$H\big(\text{Sex—Long hair}\big) = -\tfrac{4}{10} \cdot 0 - \ldots$$

Example

ID	Height	Weight	**Long hair**	Sex
1	m	n	n	m
2	s	l	y	f
3	t	h	n	m
4	s	n	y	f
5	t	n	y	f
6	s	l	n	f
7	s	h	n	m
8	m	n	n	f
9	m	l	y	f
10	t	n	n	m

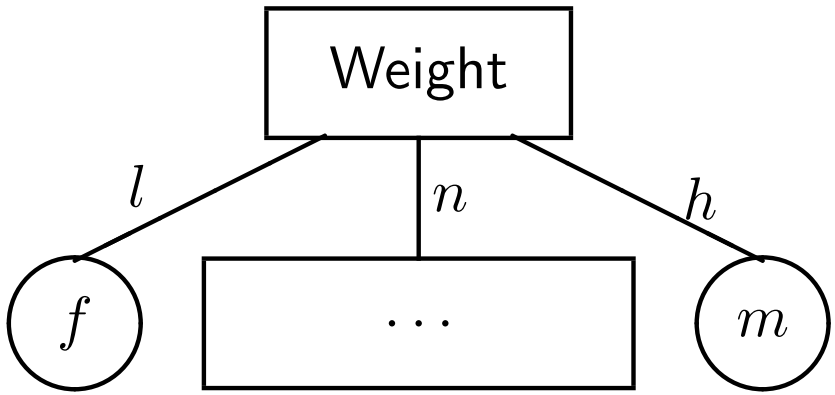
$$H(\text{Sex—Long hair}) = \ldots - \frac{6}{10}\left(\frac{2}{6}\log_2\left(\frac{2}{6}\right) + \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right)$$

Example

$$
\begin{aligned}
H(\text{Sex—Long hair}) &= -\frac{4}{10} \cdot 0 \\
&\quad -\frac{6}{10}\left(\frac{2}{6}\log_2\left(\frac{2}{6}\right) + \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right) \\
&\approx 0.5510
\end{aligned}
$$

Example

The attribute *Weight* yields the largest reduction of entropy.



Example

The remaining data table to be considered in the node $\boxed{\cdots}$:

ID	Height	Long hair	Sex
1	m	n	m
4	s	y	f
5	t	y	f
8	m	n	f
10	t	n	m

Example

ID	Height	Long hair	Sex
1	m	n	m
4	s	y	f
5	t	y	f
8	m	n	f
10	t	n	m

$$H(\text{Sex—Weight=n}) = -\left(\tfrac{2}{5}\log_2\left(\tfrac{2}{5}\right) + \tfrac{3}{5}\log_2\left(\tfrac{3}{5}\right)\right) \approx 0.9710$$

Example

ID	**Height**	Long hair	Sex
1	m	n	m
4	s	y	f
5	t	y	f
8	m	n	f
10	t	n	m

$$H\big(\text{Sex—Weight=n, Height}\big) = -\tfrac{1}{5} \cdot 0 - \ldots$$

Example

ID	**Height**	Long hair	Sex
1	m	n	m
4	s	y	f
5	t	y	f
8	m	n	f
10	t	n	m

$$H(\text{Sex—Weight=n, Height}) = \ldots - \tfrac{2}{5}\left(\tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right) + \tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right)\right) - \ldots$$

Example

ID	**Height**	Long hair	Sex
1	m	n	m
4	s	y	f
5	t	y	f
8	m	n	f
10	t	n	m

$$H(\text{Sex—Weight=n, Height}) = \ldots - \tfrac{2}{5}\left(\tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right) + \tfrac{1}{2}\log_2\left(\tfrac{1}{2}\right)\right) = 0.8$$

Example

ID	Height	**Long hair**	Sex
1	m	n	m
4	s	y	f
5	t	y	f
8	m	n	f
10	t	n	m

$$H(\text{Sex—Weight=n, Long hair}) = -\tfrac{2}{5} \cdot 0 - \ldots$$

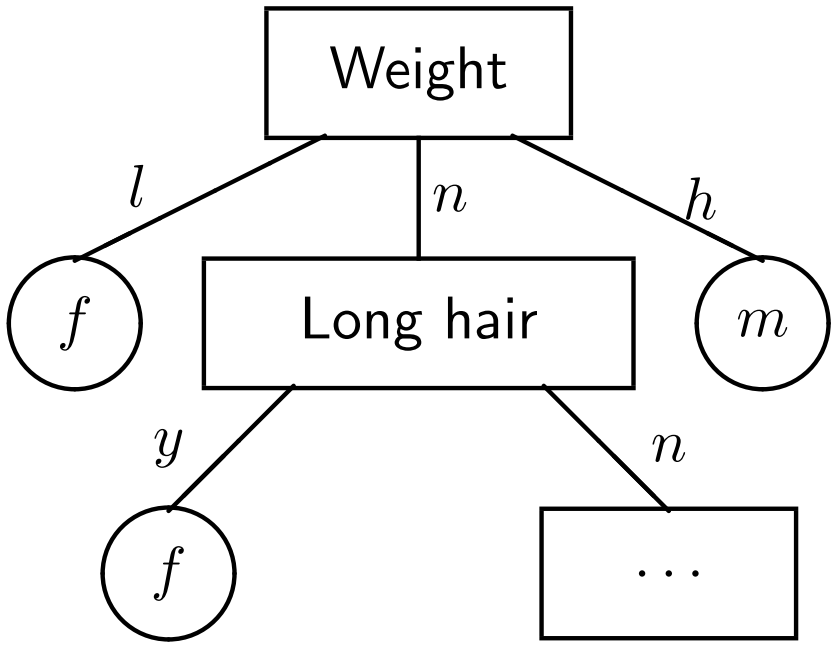
Example

ID	Height	**Long hair**	Sex
1	m	n	m
4	s	y	f
5	t	y	f
8	m	n	f
10	t	n	m

$$H(\text{Sex—Weight=n, Long hair}) = \ldots - \tfrac{3}{5}\left(\tfrac{1}{3}\log_2\left(\tfrac{1}{3}\right) + \tfrac{2}{3}\log_2\left(\tfrac{2}{3}\right)\right) \approx 0.5510$$

Example

The attribute *long hair* yields the largest reduction of entropy.



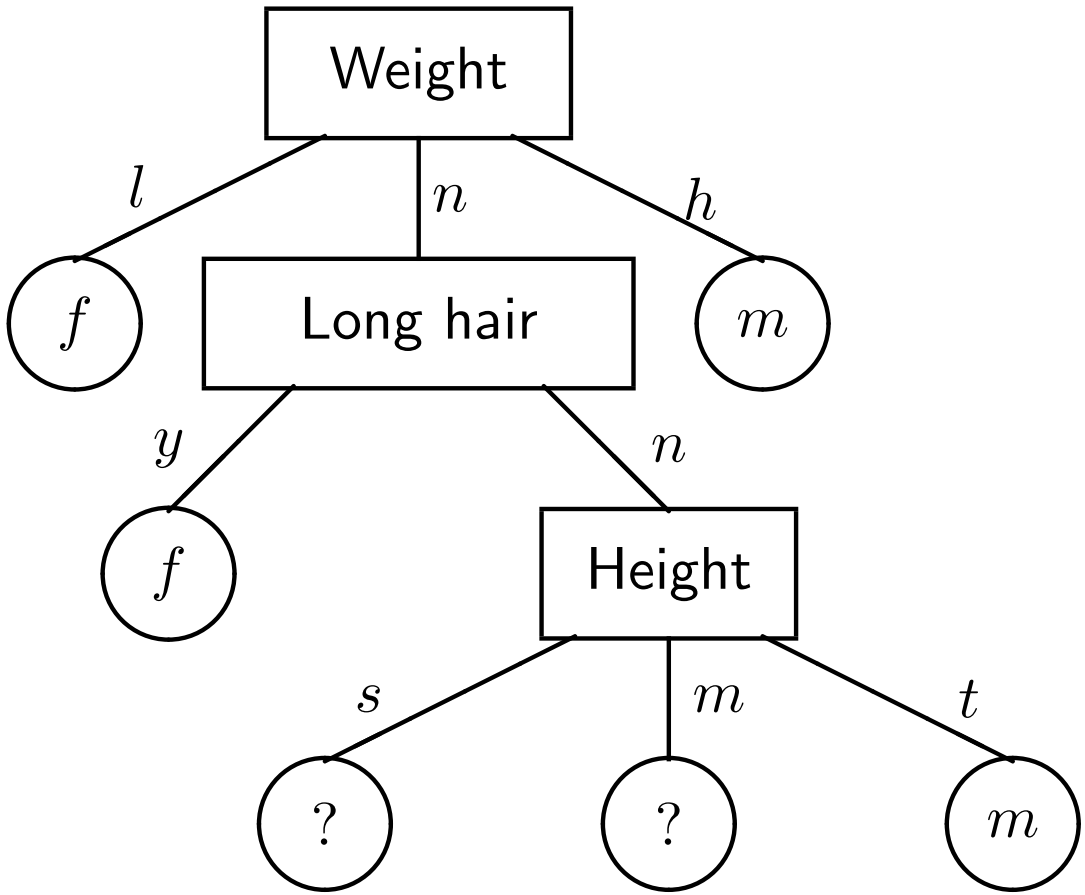
Example

For the remaining node, only the attribute *Height* is left with the remaining data table:

ID	Height	Sex
1	m	m
8	m	f
10	t	m

Therefore, the resulting decision tree is:

Example



Axioms for entropy

(H1) Entropy is a (class of) real-valued funktion(s) $H(\mathbf{p})$ which is defined for all probability distributions $\mathbf{p} \in \mathbb{R}^n$ ($\sum_{i=1}^{n} p_i = 1$ und $p_i \geq 0$ for all $i = 1, \ldots, n$) with a finite number of outcomes.

(H2) Entropy is never negative, i.e. $H(\mathbf{p}) \geq 0$ where $H(\mathbf{p}) = 0$ holds only in the case of a certain event when ($p_i = 1$ for one i).

(H3) If the probability distributions $\mathbf{p}$ and $\mathbf{q}$ are identical, except that $\mathbf{p}$ has some extra events with probability 0, then $H(\mathbf{p}) = H(\mathbf{q})$ holds. (Impossible events have no influence on the netropy.)

Axioms for entropy

(H4) $H(1/n, \ldots, 1/n)$ is increasing with n, i.e. the entropy of the uniform distribution increases with the number of possible outcomes.

(H5) $H(\mathbf{p})$ is a continuous function in $\mathbf{p}$, i.e. entropy does not change in steps.

(H6) When random experiments are concatenated, entropy can be computed by a suitable weighted sum.

Other evaluation measures from information theory

Normalized Information Gain

- Information gain is biased towards many-valued attributes i.e., of two attributes having about the same information content it tends to select the one having more values.
- Normalization removes / reduces this bias.

Information Gain Ratio (Quinlan 1986 / 1993)

$$I_{\text{gr}}(C, A) = \frac{I_{\text{gain}}(C, A)}{H(A)} = \frac{I_{\text{gain}}(C, A)}{-\sum_{j=1}^{n_A} p_{\cdot j} \log_2 p_{\cdot j}}$$

Gini index

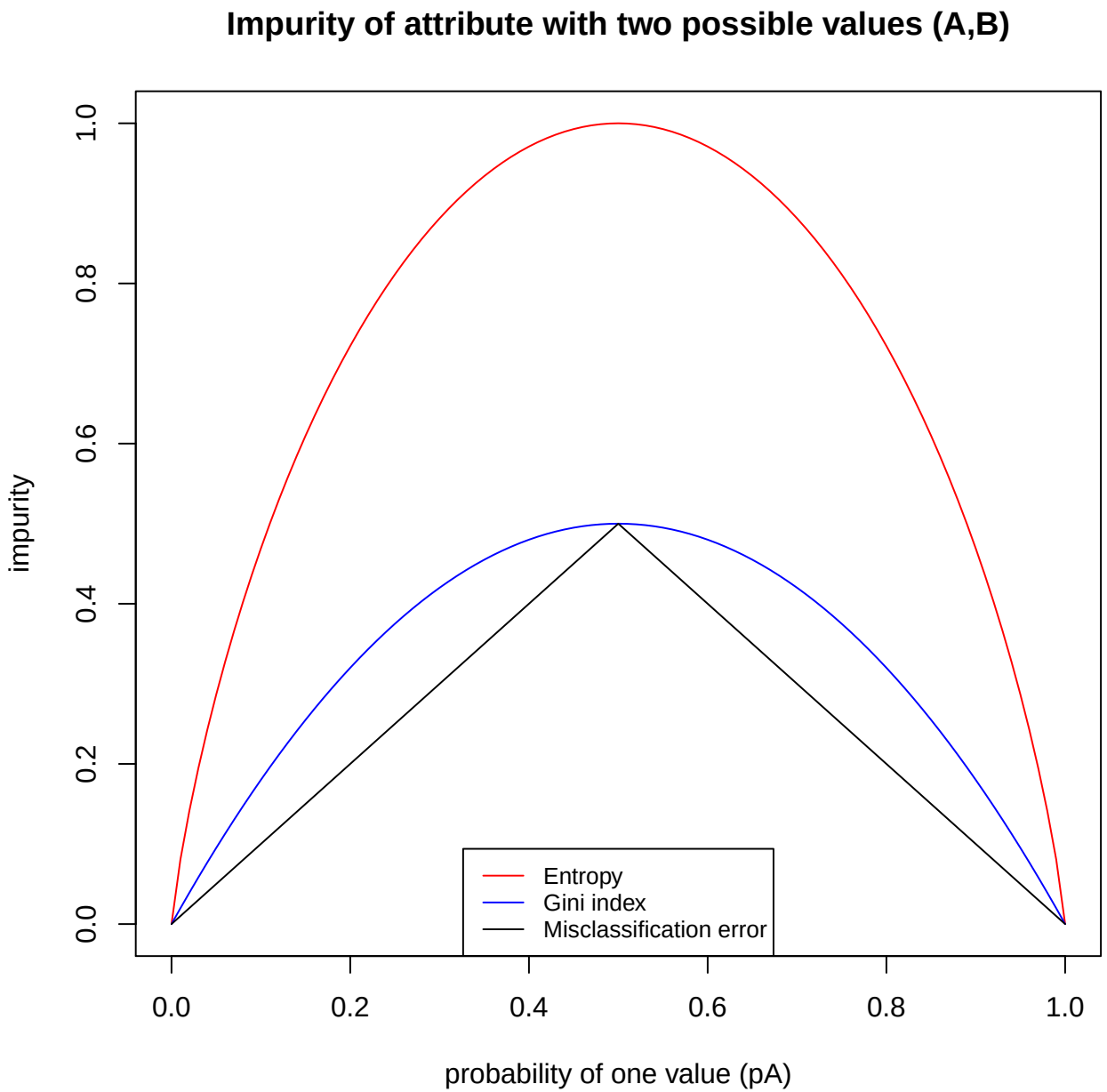
- Rate of wrong classifications of a random data point (training set) on the basis of the distribution of the labels (training set).
- Can be interpreted as expected error rate

$$I_{Gini}(C) = 1 - \sum_{i=1}^{n_C} p_i^2$$

$$I_{\text{GiniGain}}(C, A) = I_{Gini}(C) - I_{Gini}(C|A)$$

$$= \overbrace{1 - \sum_{i=1}^{n_C} p_i^2}^{I_{Gini}(C)} - \overbrace{\sum_{j=1}^{n_A} p_{.j} \left(1 - \sum_{i=1}^{n_C} p_{i|j}^2\right)}^{I_{Gini}(C|A)}$$

Comparison of impurity measures



χ^2-measure

- Compares the actual joint distribution with a **hypothetical independent distribution**.
- Uses absolute comparison.
- Can be interpreted as a difference measure.

$$\chi^2(C, A) = \sum_{i=1}^{n_C} \sum_{j=1}^{n_A} N_{..} \frac{(p_{i.} p_{.j} - p_{ij})^2}{p_{i.} p_{.j}}$$

Contingency tables

$X \setminus Y$	y_1	$\dots$	y_j	$\dots$	y_q	marginal of X
x_1	p_{11}	$\dots$	p_{1j}	$\dots$	p_{1q}	$p_{1\bullet}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_i	p_{i1}	$\dots$	p_{ij}	$\dots$	p_{iq}	$p_{i\bullet}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_r	p_{r1}	$\dots$	p_{rj}	$\dots$	p_{rq}	$p_{r\bullet}$
marginal of Y	$p_{\bullet 1}$	$\dots$	$p_{\bullet j}$	$\dots$	$p_{\bullet q}$	n

The random variable X can take the values $x_1, \dots, x_r$, the random variable Y the values $y_1, \dots, y_q$.

p_{ij} is the (absolute) frequency of occurrences of the observation (x_i, y_j).

Contingency tables

$$p_{i\bullet} = \sum_{j=1}^{q} p_{ij} \quad \text{and} \quad p_{\bullet j} = \sum_{i=1}^{r} p_{ij}$$

are the marginal (absolute) frequencies.

If X and Y are independent, then the expected absolute frequencies are

$$e_{ij} = \frac{p_{i\bullet} p_{\bullet j}}{n}$$

for all $i \in \{1, \ldots, r\}$ and all $j \in \{1, \ldots, q\}$.

χ^2 independence test

Example. 1000 people were asked which political party they voted for in order to find out whether the choice of the party and the sex of the voter are independent.

pol. party\ sex	female	male	sum
SPD	200	170	370
CDU/CSU	200	200	400
Grüne	45	35	80
FDP	25	35	70
PDS	20	30	50
Others	22	5	27
No answer	8	5	13
sum	520	480	1000

χ^2 independence test

Expected frequencies:

pol. party\ sex	female	male
SPD	192.4	177.6
CDU/CSU	208.0	192.0
Grüne	41.6	38.4
FDP	31.2	28.4
PDS	26.0	24.0
O/NA	20.8	19.2

χ^2 independence test

pol. party\ sex	female	male	sum
SPD	200	170	370
CDU/CSU	200	200	400
Grüne	45	35	80
FDP	25	35	70
PDS	20	30	50
O/NA	30	10	40
sum	520	480	1000

For instance: $e_{\text{CDU/CSU, female}} = \frac{400}{1000} \cdot \frac{520}{1000} \cdot 1000 = 208.0$

Treatment of numerical attributes

General Approach: Discretization

- **Preprocessing I**
  - Form equally sized or equally populated intervals (binning).
- **Preprocessing II / Multisplits during tree construction**
  - Build a decision tree using only the numeric attribute.
  - Flatten the tree to obtain a multi-interval discretization.

Treatment of numerical attributes

Splits at boundary points minimize entropy.

The boundary points are marked by lines.

Value:	1	2	3	3	4	5	5	6	6
Class:	c	c	a	a	a	b	b	a	c

7	8	8	9	10	11	11	12
c	c	c	c	b	a	a	a

- For binary splits (only one cut point) all boundary points are considered and the one with the smallest entropy is chosen.
- For multiple splits a recursive procedure is applied.

Treatment of missing values

Induction

- Weight the evaluation measure with the fraction of cases with known values.
  - Idea: The attribute provides information only if it is known.
- Try to find a surrogate test attribute with similar properties (CART, Breiman *et al.* 1984)
- Assign the case to all branches, weighted in each branch with the relative frequency of the corresponding attribute value (C4.5, Quinlan 1993).

Treatment of missing values

Classification

- Use the surrogate test attribute found during induction.
- Follow all branches of the test attribute, weighted with their relative number of cases, aggregate the class distributions of all leaves reached, and assign the majority class of the aggregated class distribution.

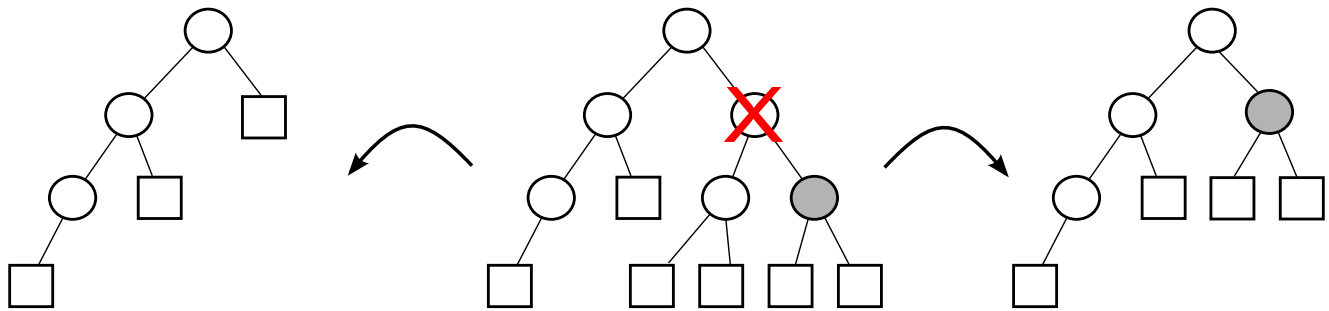
Pruning decision trees

Pruning serves the purpose

- to simplify the tree (improve interpretability),
- to avoid overfitting (improve generalization).

Basic ideas:

- Replace "bad" branches (subtrees) by leaves.
- Replace a subtree by its largest branch if it is better.



Pruning decision trees

Common approaches:

- Reduced error pruning
- Pessimistic pruning
- Confidence level pruning

Reduced error pruning

- Classify a set of new example cases with the decision tree. (These cases must not have been used for the induction!)
- Determine the number of errors for all leaves.
- The number of errors of a subtree is the sum of the errors of all of its leaves.
- Determine the number of errors for leaves that replace subtrees.
- If such a leaf leads to the same or fewer errors than the subtree, replace the subtree by the leaf.
- If a subtree has been replaced, recompute the number of errors of the subtrees it is part of.

Reduced error pruning

Advantage: Very good pruning, effective avoidance of overfitting.

Disadvantage: Additional example cases needed.

Number of cases in a leaf has no influence.

Pessimistic pruning

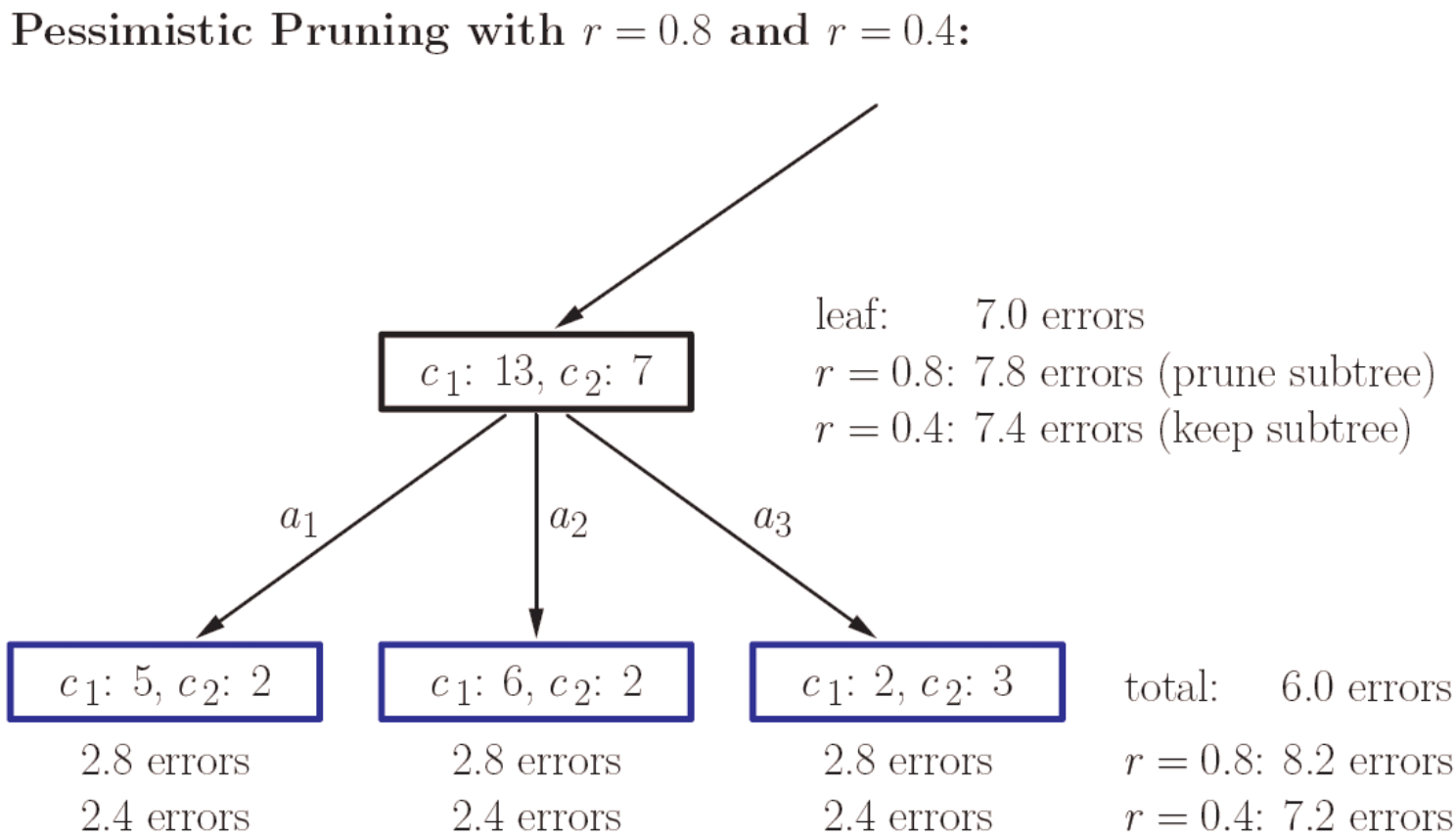
- Classify a set of example cases with the decision tree. (These cases may or may not have been used for the induction.)
- Determine the number of errors for all leaves and increase this number by a fixed, user-specified amount r.
- The number of errors of a subtree is the sum of the errors of all of its leaves.
- Determine the number of errors for leaves that replace subtrees (also increased by r).
- If such a leaf leads to the same or fewer errors than the subtree, replace the subtree by the leaf and recompute subtree errors.

Pessimistic pruning

Advantage: No additional example cases needed.

Disadvantage: Number of cases in a leaf has no influence.

Pessimistic pruning: An example



Confidence level pruning

Like pessimistic pruning, but the number of errors is computed as follows:

- See classification in a leaf as a Bernoulli experiment (error/no error): p, $p(1-p)$
  - Expected success rate: $f = \frac{\text{no error}}{\text{error + no error}}$
  - For a large enough number of classifications f follows a normal distribution
- Estimate an interval for the error probability $p(1-p)$ based on a user-specified confidence level α. (use approximation of the binomial distribution by a normal distribution)
- Increase error number to the upper level of the confidence interval times the number of cases assigned to the leaf.
- Formal problem: Classification is not a random experiment.

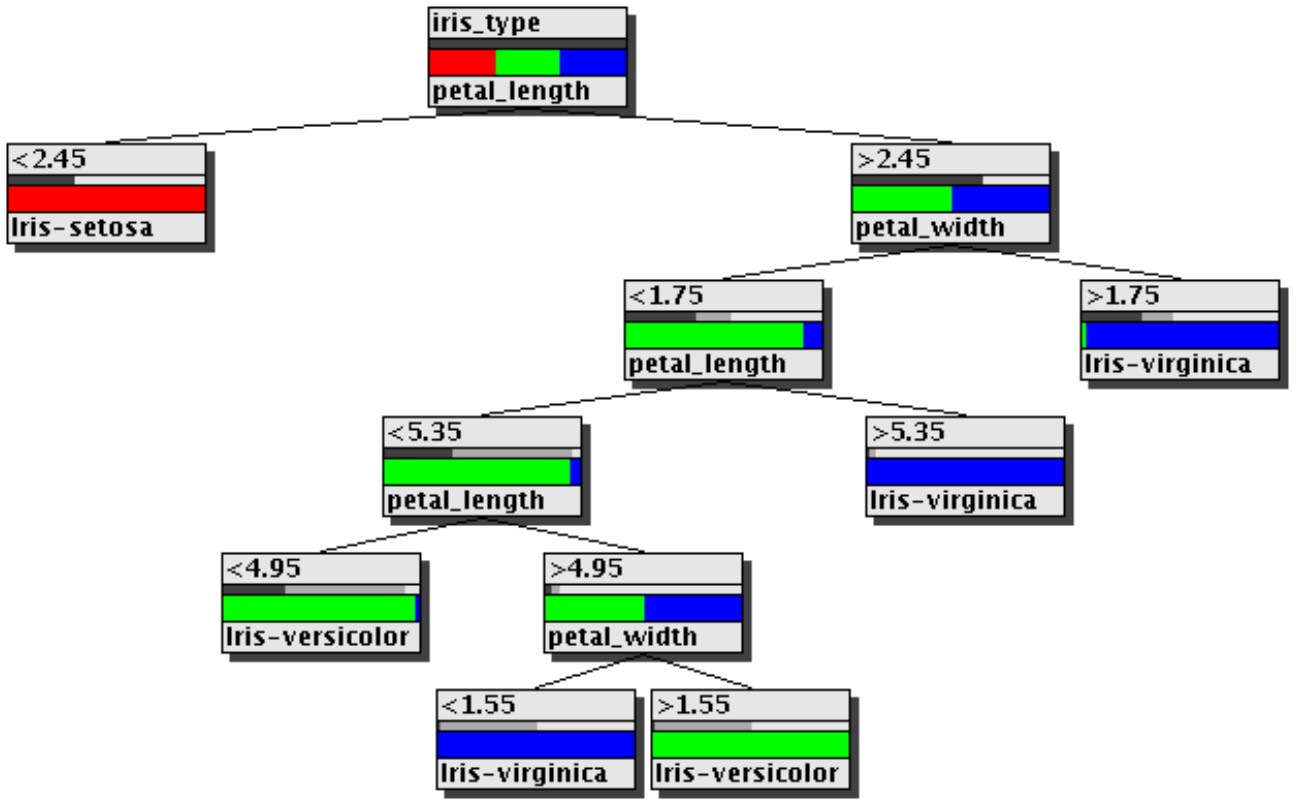
Confidence level pruning

Advantage: No additional example cases needed, good pruning.

Disadvantage: Statistically dubious foundation.

Decision tree pruning: An example

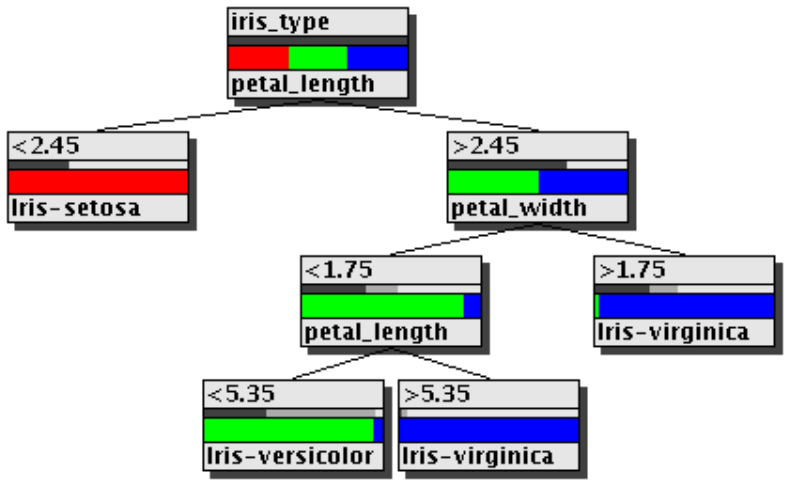
A decision tree for the Iris data
(induced with information gain ratio, unpruned)

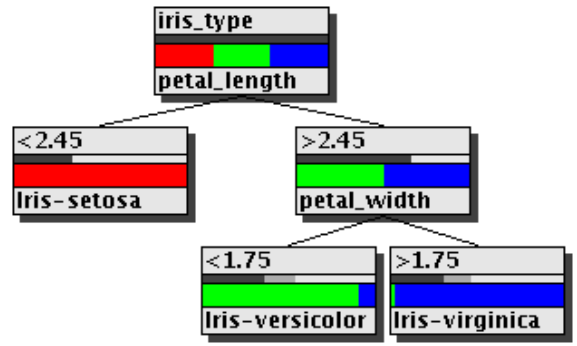


Decision tree pruning: An example

A decision tree for the Iris data

(pruned with confidence level pruning, $\alpha = 0.8$, and pessimistic pruning, $r = 2$)





- Left: 7 instead of 11 nodes, 4 instead of 2 misclassifications.
- Right: 5 instead of 11 nodes, 6 instead of 2 misclassifications.
- The right tree is "minimal" for the three classes.

Predictive vs. descriptive tasks

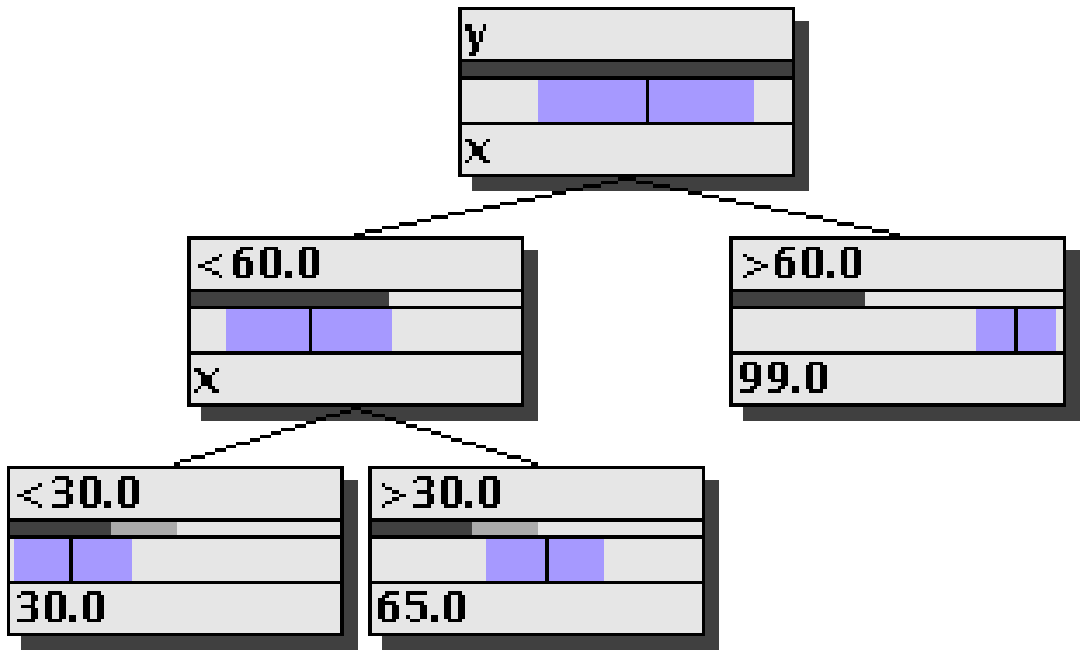
Predictive tasks: The decision tree (or more generally, the classifier) is constructed in order to apply it to new unclassified data.

Decriptive tasks: The purpose of the tree construction is to understand, how classification has been carried out so far.

Regression trees

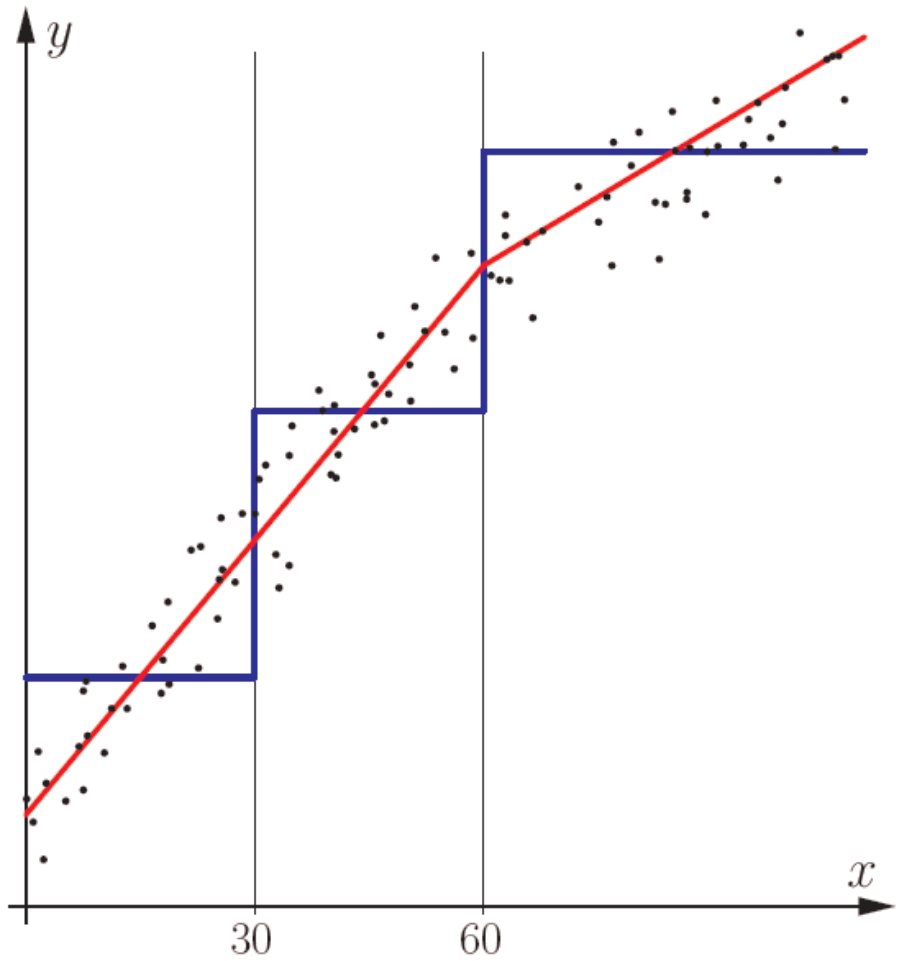
Like decision trees, but target variable is not a class, but a numeric quantity.

- Simple regression trees: Predict constant values in leaves. (blue line)

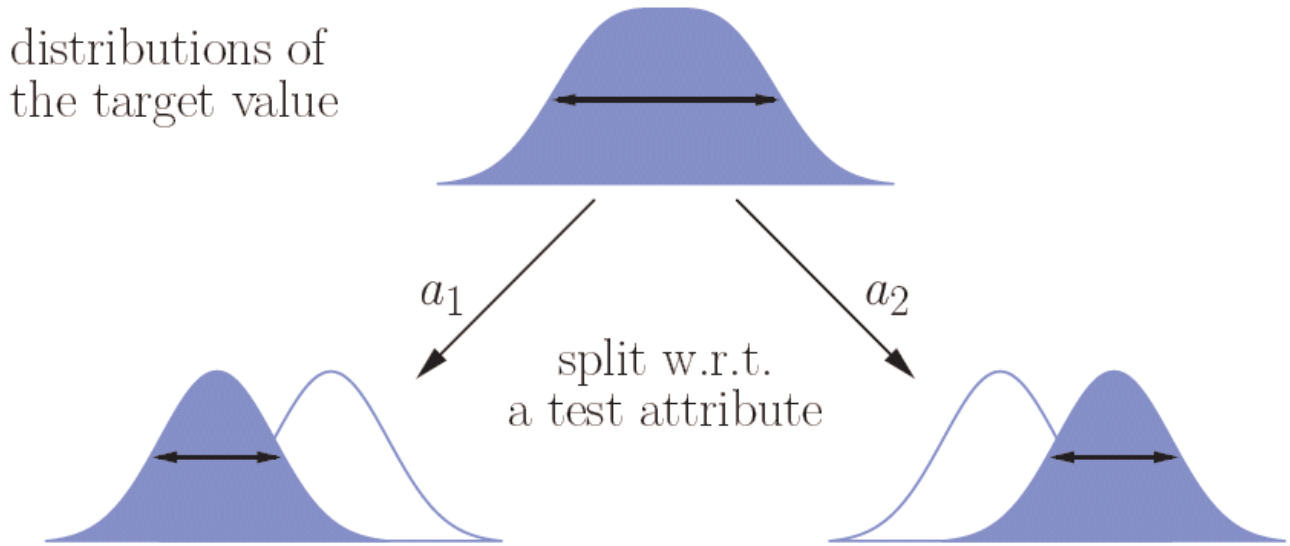


Regression trees

- More complex regression trees: Predict linear functions in leaves. (red line)



Regression trees: Attribute selection



- The variance/standard deviation is compared to the variance/standard deviation in the branches.
- The attribute that yields the highest reduction is selected.