Aalto University
School of Electrical Engineering

CS-C3240 – Machine Learning D

Non-Parametric methods

Stephan Sigg

Department of Communications and Networking
Aalto University, School of Electrical Engineering
stephan.sigg@aalto.fi

Version 1.0, February 7, 2022

Learning goals

Understand the concepts of

- Decision trees
- Information score
- Estimation of error rates
- Pruning

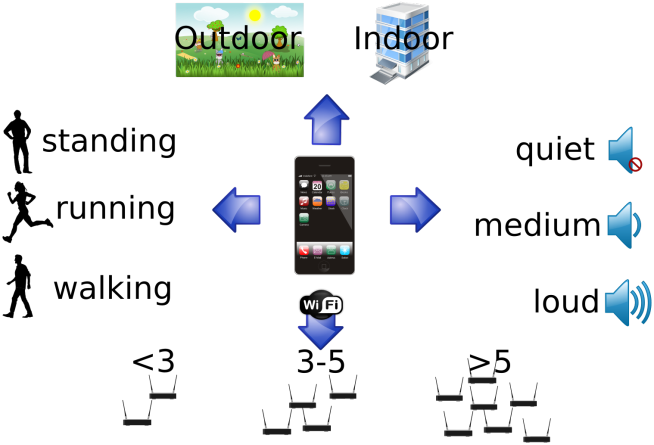
Outline

Decision Trees

Optimizing the tree structure

Improving classification results

Smartphone sensing:
At work or not ?



Decision trees

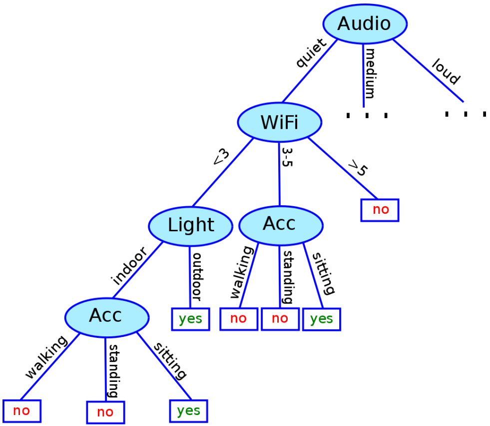
Assume that some training data was recorded and labelled for the two classes we consider in this example

#	WiFi	ACC	Audio	Light	Label
1	*<3*	*walking*	*quiet*	*outdoor*	Work
2	*<3*	*walking*	*quiet*	*outdoor*	Work
3	*3-5*	*walking*	*quiet*	*outdoor*	Work
4	*3-5*	*sitting*	*quiet*	*outdoor*	Work
5	*<3*	*sitting*	*quiet*	*indoor*	Work
6	*3-5*	*sitting*	*quiet*	*indoor*	Work
7	*3-5*	*sitting*	*quiet*	*indoor*	Work
8	*3-5*	*sitting*	*quiet*	*indoor*	Work
9	*>5*	*walking*	*loud*	*indoor*	Work
10	*>5*	*standing*	*medium*	*indoor*	Work
11	*>5*	*sitting*	*medium*	*indoor*	Work
12	*>5*	*sitting*	*medium*	*indoor*	Work
13	*>5*	*sitting*	*medium*	*indoor*	Work
14	*>5*	*sitting*	*medium*	*indoor*	Work
15	*>5*	*sitting*	*medium*	*indoor*	Work
16	*>5*	*sitting*	*loud*	*indoor*	Work
17	*<3*	*walking*	*quiet*	*indoor*	Not at work
18	*<3*	*walking*	*quiet*	*indoor*	Not at work
19	*<3*	*standing*	*quiet*	*indoor*	Not at work
20	*<3*	*walking*	*medium*	*indoor*	Not at work
21	*<3*	*walking*	*loud*	*outdoor*	Not at work
22	*<3*	*walking*	*medium*	*indoor*	Not at work
23	*<3*	*walking*	*medium*	*indoor*	Not at work
24	*3-5*	*walking*	*quiet*	*outdoor*	Not at work
25	*3-5*	*standing*	*quiet*	*outdoor*	Not at work
26	*3-5*	*standing*	*quiet*	*outdoor*	Not at work
27	*3-5*	*standing*	*loud*	*outdoor*	Not at work
28	*3-5*	*walking*	*loud*	*outdoor*	Not at work
29	*>5*	*sitting*	*loud*	*outdoor*	Not at work
30	*>5*	*sitting*	*loud*	*outdoor*	Not at work

Decision trees

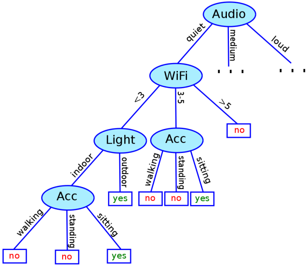
A decision tree divides the examples from a dataset according to the features and classes observed for them

#	WiFi	ACC	Audio	Light	Label
1	<3	walking	quiet	outdoor	Work
2	<3	walking	quiet	outdoor	Work
3	3-5	walking	quiet	outdoor	Work
4	3-5	sitting	quiet	outdoor	Work
5	<3	sitting	quiet	indoor	Work
6	3-5	sitting	quiet	indoor	Work
7	3-5	sitting	quiet	indoor	Work
8	3-5	sitting	quiet	indoor	Work
9	>5	walking	loud	indoor	Work
10	>5	standing	medium	indoor	Work
11	>5	sitting	medium	indoor	Work
12	>5	sitting	medium	indoor	Work
13	>5	sitting	medium	indoor	Work
14	>5	sitting	medium	indoor	Work
15	>5	sitting	medium	indoor	Work
16	>5	sitting	loud	indoor	Work
17	<3	walking	quiet	indoor	Not at work
18	<3	walking	quiet	indoor	Not at work
19	<3	standing	quiet	indoor	Not at work
20	<3	walking	medium	indoor	Not at work
21	<3	walking	loud	outdoor	Not at work
22	<3	walking	medium	indoor	Not at work
23	<3	walking	medium	indoor	Not at work
24	3-5	walking	quiet	outdoor	Not at work
25	3-5	standing	quiet	outdoor	Not at work
26	3-5	standing	quiet	outdoor	Not at work
27	3-5	standing	loud	outdoor	Not at work
28	3-5	walking	loud	outdoor	Not at work
29	>5	sitting	loud	outdoor	Not at work
30	>5	sitting	loud	outdoor	Not at work



Decision tree

How to generate such decision tree?

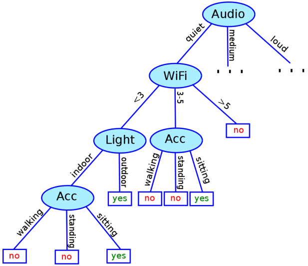


Decision tree

How to generate such decision tree?

First select a feature to split on and place it at the root node.

Then repeat this procedure for all child nodes



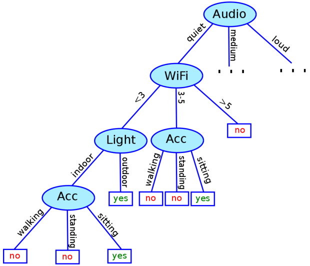
Decision tree

How to generate such decision tree?

First select a feature to split on and place it at the root node.

Then repeat this procedure for all child nodes

How to determine the feature to split on?



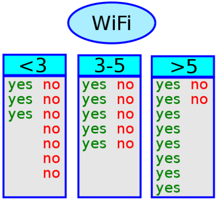
Decision tree

WiFi	yes	no	Accelerometer	yes	no	Audio	yes	no	Light	yes	no	At work yes	At work no
<3 APs	3	7	walking	4	8	quiet	8	5	outdoor	4	7	16	14
[3, 5]	5	5	standing	1	4	medium	6	3	indoor	12	7		
>5 APs	8	2	sitting	11	2	loud	2	6					

#	WiFi	ACC	Audio	Light	Label
1	*<3*	*walking*	*quiet*	*outdoor*	Work
2	*<3*	*walking*	*quiet*	*outdoor*	Work
3	*3-5*	*walking*	*quiet*	*outdoor*	Work
4	*3-5*	*sitting*	*quiet*	*outdoor*	Work
5	*<3*	*sitting*	*quiet*	*indoor*	Work
6	*3-5*	*sitting*	*quiet*	*indoor*	Work
7	*3-5*	*sitting*	*quiet*	*indoor*	Work
8	*3-5*	*sitting*	*quiet*	*indoor*	Work
9	*>5*	*walking*	*loud*	*indoor*	Work
10	*>5*	*standing*	*medium*	*indoor*	Work
11	*>5*	*sitting*	*medium*	*indoor*	Work
12	*>5*	*sitting*	*medium*	*indoor*	Work
13	*>5*	*sitting*	*medium*	*indoor*	Work
14	*>5*	*sitting*	*medium*	*indoor*	Work
15	*>5*	*sitting*	*medium*	*indoor*	Work
16	*>5*	*sitting*	*loud*	*indoor*	Work
17	*<3*	*walking*	*quiet*	*indoor*	Not at work
18	*<3*	*walking*	*quiet*	*indoor*	Not at work
19	*<3*	*standing*	*quiet*	*indoor*	Not at work
20	*<3*	*walking*	*medium*	*indoor*	Not at work
21	*<3*	*walking*	*loud*	*outdoor*	Not at work
22	*<3*	*walking*	*medium*	*indoor*	Not at work
23	*<3*	*walking*	*medium*	*indoor*	Not at work
24	*3-5*	*walking*	*quiet*	*outdoor*	Not at work
25	*3-5*	*standing*	*quiet*	*outdoor*	Not at work
26	*3-5*	*standing*	*quiet*	*outdoor*	Not at work
27	*3-5*	*standing*	*loud*	*outdoor*	Not at work
28	*3-5*	*walking*	*loud*	*outdoor*	Not at work
29	*>5*	*sitting*	*loud*	*outdoor*	Not at work
30	*>5*	*sitting*	*loud*	*outdoor*	Not at work

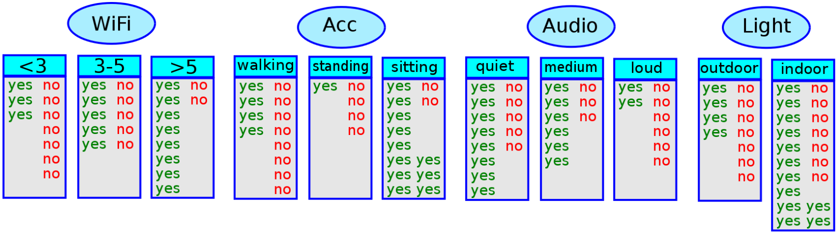
Decision tree

WiFi			Accelerometer			Audio			Light			At work	
	yes	no		yes	no		yes	no		yes	no	yes	no
$<$3 APs	3	7	walking	4	8	quiet	8	5	outdoor	4	7	16	14
[3, 5]	5	5	standing	1	4	medium	6	3	indoor	12	7		
$>$5 APs	8	2	sitting	11	2	loud	2	6					



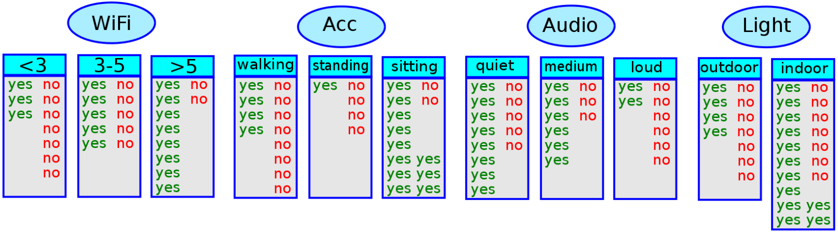
Decision tree

WiFi			Accelerometer			Audio			Light			At work	
	yes	no		yes	no		yes	no		yes	no	yes	no
<3 APs	3	7	walking	4	8	quiet	8	5	outdoor	4	7	16	14
[3, 5]	5	5	standing	1	4	medium	6	3	indoor	12	7		
>5 APs	8	2	sitting	11	2	loud	2	6					



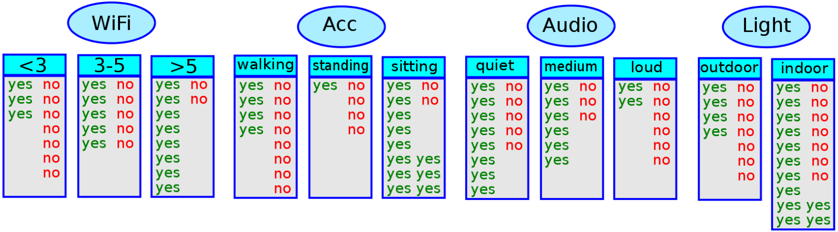
Which feature is the best choice to place at the root?

Decision tree



We are interested in the gain in information when a particular choice is taken

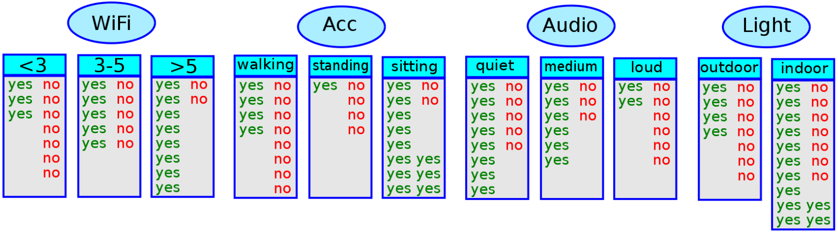
Decision tree



We are interested in the gain in information when a particular choice is taken

The decision tree should decide for the split that promises maximum information gain.

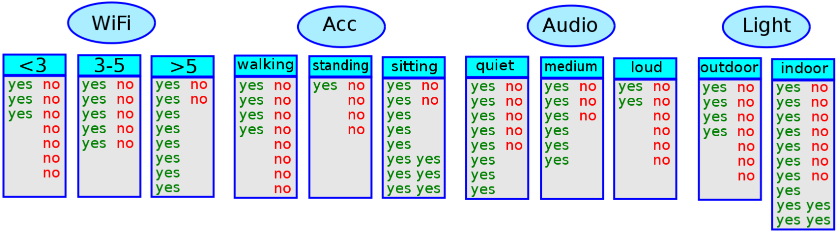
Decision tree



Information gain can be estimated by the entropy of a value:

$$\mathcal{E}(p_1, p_2, \ldots, p_n) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \cdots - p_n \log_2 p_n$$

Decision tree

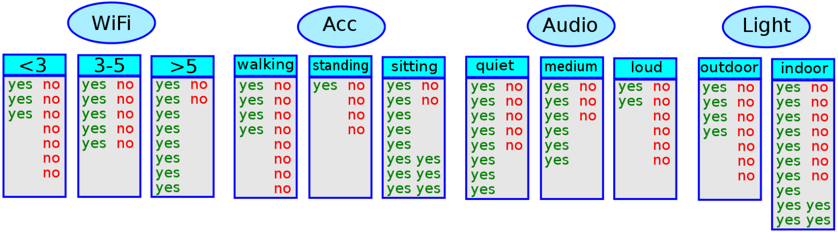


$$\mathcal{E}(p_1, p_2, \ldots, p_n) = \quad -p_1 \log_2 p_1 - p_2 \log_2 p_2 \cdots - p_n \log_2 p_n$$

WiFi information value:

$$\mathcal{E}\left(\frac{3}{10}, \frac{7}{10}\right)\frac{10}{30} + \mathcal{E}\left(\frac{5}{10}, \frac{5}{10}\right)\frac{10}{30} + \mathcal{E}\left(\frac{8}{10}, \frac{2}{10}\right)\frac{10}{30} =$$

Decision tree

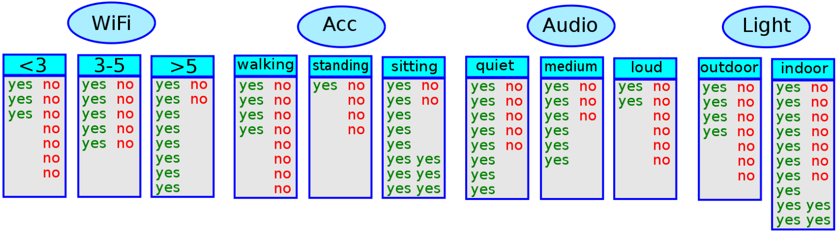


$$\mathcal{E}(p_1, p_2, \dots, p_n) = \quad -p_1 \log_2 p_1 - p_2 \log_2 p_2 \cdots - p_n \log_2 p_n$$

WiFi information value:

$$\mathcal{E}\left(\frac{3}{10}, \frac{7}{10}\right)\frac{10}{30} + \mathcal{E}\left(\frac{5}{10}, \frac{5}{10}\right)\frac{10}{30} + \mathcal{E}\left(\frac{8}{10}, \frac{2}{10}\right)\frac{10}{30} = \left(-\tfrac{3}{10}\log_2\tfrac{3}{10} - \tfrac{7}{10}\log_2\tfrac{7}{10}\right)\cdot\tfrac{10}{30}$$
$$+\left(-\tfrac{5}{10}\log_2\tfrac{5}{10} - \tfrac{5}{10}\log_2\tfrac{5}{10}\right)\cdot\tfrac{10}{30}$$
$$+\left(-\tfrac{8}{10}\log_2\tfrac{8}{10} - \tfrac{2}{10}\log_2\tfrac{2}{10}\right)\cdot\tfrac{10}{30}$$

Decision tree

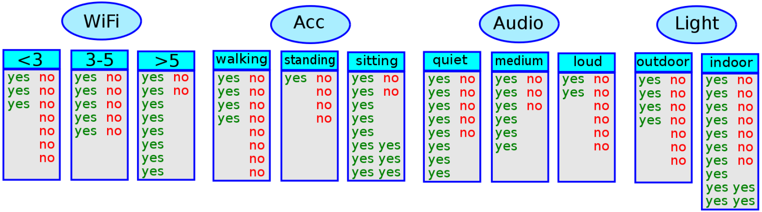


$$\mathcal{E}(p_1, p_2, \ldots, p_n) = \quad -p_1 \log_2 p_1 - p_2 \log_2 p_2 \cdots - p_n \log_2 p_n$$

WiFi information value:

$$\begin{aligned}
\mathcal{E}\left(\frac{3}{10}, \frac{7}{10}\right)\frac{10}{30} + \mathcal{E}\left(\frac{5}{10}, \frac{5}{10}\right)\frac{10}{30} + \mathcal{E}\left(\frac{8}{10}, \frac{2}{10}\right)\frac{10}{30} = & \quad \left(-\tfrac{3}{10}\log_2\tfrac{3}{10} - \tfrac{7}{10}\log_2\tfrac{7}{10}\right)\cdot\tfrac{10}{30} \\
& + \left(-\tfrac{5}{10}\log_2\tfrac{5}{10} - \tfrac{5}{10}\log_2\tfrac{5}{10}\right)\cdot\tfrac{10}{30} \\
& + \left(-\tfrac{8}{10}\log_2\tfrac{8}{10} - \tfrac{2}{10}\log_2\tfrac{2}{10}\right)\cdot\tfrac{10}{30} \\
\approx & \quad 0.868
\end{aligned}$$

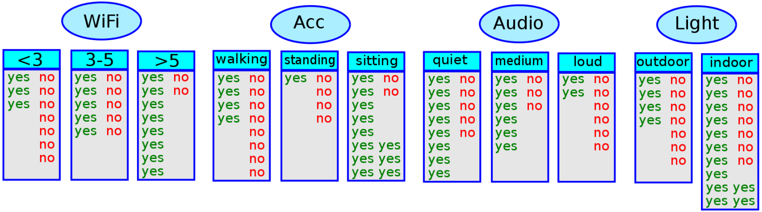
Decision tree



Information value:

WiFi:	$\approx$	0.868
Acc:	$\approx$	...
Audio:	$\approx$	...
Light:	$\approx$	...

Decision tree



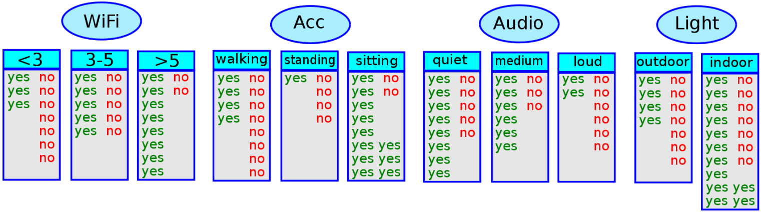
Information value:

Information gain:

WiFi:	$\approx$	0.868
Acc:	$\approx$	0.756
Audio:	$\approx$	0.884
Light:	$\approx$	0.948

Initial information value (working [yes/no]): 0.997

Decision tree



Information value:

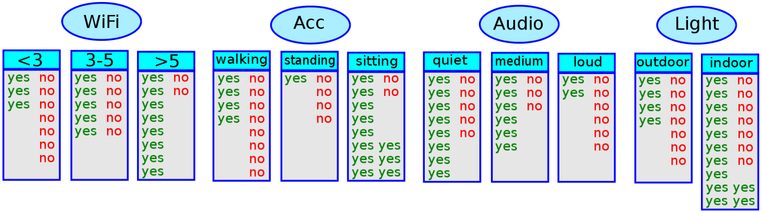
WiFi: $\approx$ 0.868
Acc: $\approx$ 0.756
Audio: $\approx$ 0.884
Light: $\approx$ 0.948

Information gain:

WiFi: $\approx$ 0.129
Acc: $\approx$ 0.241
Audio: $\approx$ 0.113
Light: $\approx$ 0.049

Initial information value (working [yes/no]): 0.997

Decision tree



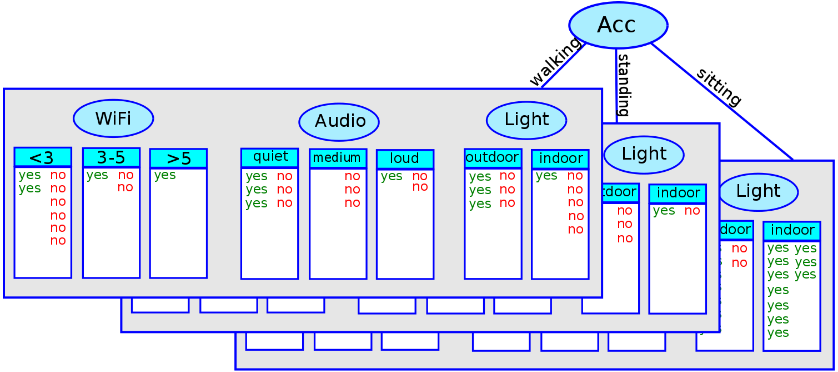
Information value:

WiFi:	$\approx$	0.868
Acc:	$\approx$	**0.756**
Audio:	$\approx$	0.884
Light:	$\approx$	0.948

Information gain:

WiFi:	$\approx$	0.129
Acc:	$\approx$	**0.241**
Audio:	$\approx$	0.113
Light:	$\approx$	0.049

Initial information value (working [yes/no]): 0.997



walking

#	WiFi	ACC	Audio	Light	Label
1	<3	walking	quiet	outdoor	Work
2	<3	walking	quiet	outdoor	Work
3	3-5	walking	quiet	outdoor	Work
9	>5	walking	loud	indoor	Work
17	<3	walking	quiet	indoor	Not at work
18	<3	walking	quiet	indoor	Not at work
20	<3	walking	medium	indoor	Not at work
21	<3	walking	loud	outdoor	Not at work
22	<3	walking	medium	indoor	Not at work
23	<3	walking	medium	indoor	Not at work
24	3-5	walking	quiet	outdoor	Not at work
28	3-5	walking	loud	outdoor	Not at work

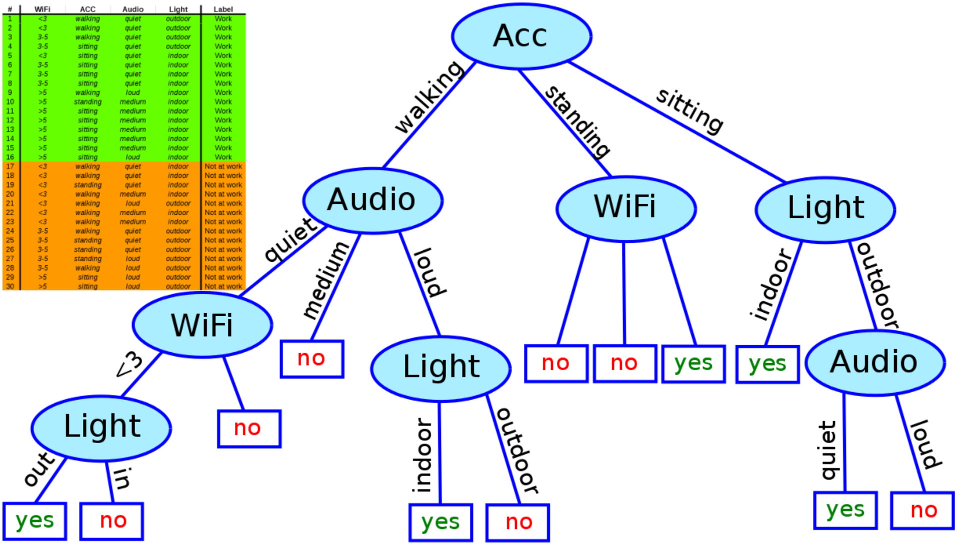
standing

#	WiFi	ACC	Audio	Light	Label
10	>5	standing	medium	indoor	Work
19	<3	standing	quiet	indoor	Not at work
25	3-5	standing	quiet	outdoor	Not at work
26	3-5	standing	quiet	outdoor	Not at work
27	3-5	standing	loud	outdoor	Not at work

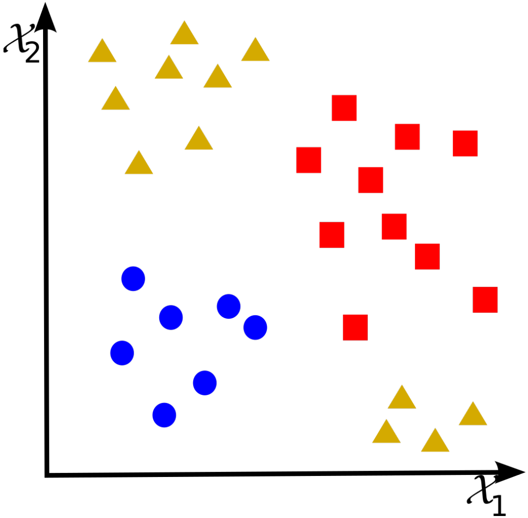
sitting

#	WiFi	ACC	Audio	Light	Label
4	3-5	sitting	quiet	outdoor	Work
5	<3	sitting	quiet	indoor	Work
6	3-5	sitting	quiet	indoor	Work
7	3-5	sitting	quiet	indoor	Work
8	3-5	sitting	quiet	indoor	Work
11	>5	sitting	medium	indoor	Work
12	>5	sitting	medium	indoor	Work
13	>5	sitting	medium	indoor	Work
14	>5	sitting	medium	indoor	Work
15	>5	sitting	medium	indoor	Work
16	>5	sitting	loud	indoor	Work
29	>5	sitting	loud	outdoor	Not at work
30	>5	sitting	loud	outdoor	Not at work

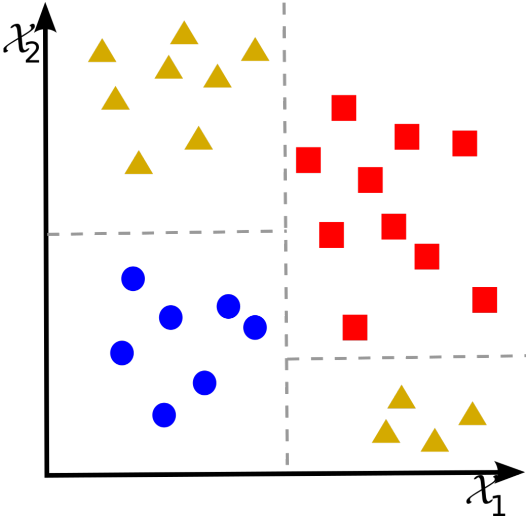
#	WiFi	ACC	Audio	Light	Label
1	<3	*walking*	*quiet*	*outdoor*	Work
2	<3	*walking*	*quiet*	*outdoor*	Work
3	3-5	*walking*	*quiet*	*outdoor*	Work
4	3-5	*sitting*	*quiet*	*outdoor*	Work
5	<3	*sitting*	*quiet*	*indoor*	Work
6	3-5	*sitting*	*quiet*	*indoor*	Work
7	3-5	*sitting*	*quiet*	*indoor*	Work
8	3-5	*sitting*	*quiet*	*indoor*	Work
9	>5	*walking*	*loud*	*indoor*	Work
10	>5	*standing*	*medium*	*indoor*	Work
11	>5	*sitting*	*medium*	*indoor*	Work
12	>5	*sitting*	*medium*	*indoor*	Work
13	>5	*sitting*	*medium*	*indoor*	Work
14	>5	*sitting*	*medium*	*indoor*	Work
15	>5	*sitting*	*medium*	*indoor*	Work
16	>5	*sitting*	*loud*	*indoor*	Work
17	<3	*walking*	*quiet*	*indoor*	Not at work
18	<3	*walking*	*quiet*	*indoor*	Not at work
19	<3	*standing*	*quiet*	*indoor*	Not at work
20	<3	*walking*	*medium*	*indoor*	Not at work
21	<3	*walking*	*loud*	*outdoor*	Not at work
22	<3	*walking*	*medium*	*indoor*	Not at work
23	<3	*walking*	*medium*	*indoor*	Not at work
24	3-5	*walking*	*quiet*	*outdoor*	Not at work
25	3-5	*standing*	*quiet*	*outdoor*	Not at work
26	3-5	*standing*	*quiet*	*outdoor*	Not at work
27	3-5	*standing*	*loud*	*outdoor*	Not at work
28	3-5	*walking*	*loud*	*outdoor*	Not at work
29	>5	*sitting*	*loud*	*outdoor*	Not at work
30	>5	*sitting*	*loud*	*outdoor*	Not at work



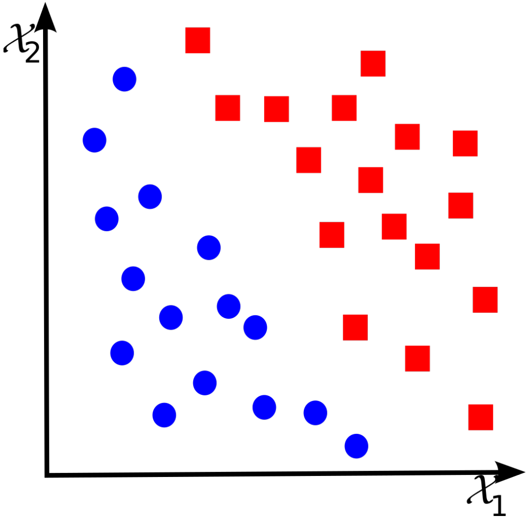
Graphical interpretation: Decision tree



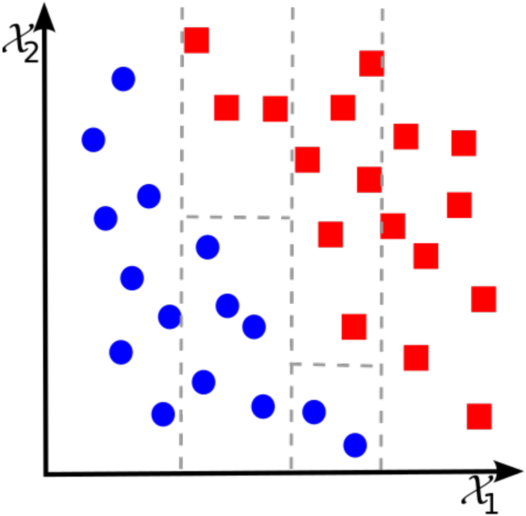
Graphical interpretation: Decision tree



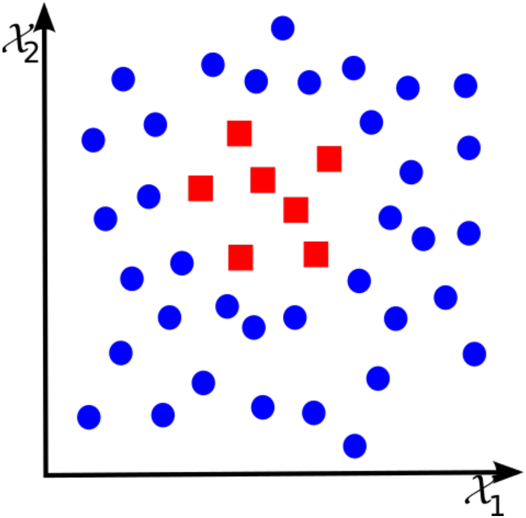
Graphical interpretation: Decision tree



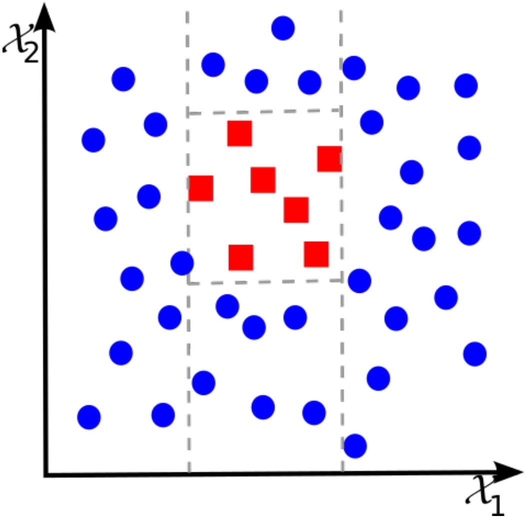
Graphical interpretation: Decision tree



Graphical interpretation: Decision tree



Graphical interpretation: Decision tree



Remark: An alternative to Information gain

Gini impurity

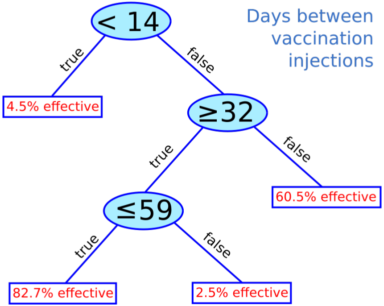
Gini impurity describes how often samples would be incorrectly labelled if labelled randomly according to the disctribution of labels in the subset. Let p_i be the probability that a sample is correctly labelled. Gini impurity is then computed as

$$I_G = \sum_{i=1}^{n} p_i \cdot (1 - p_i)$$

Regression trees

Regression trees

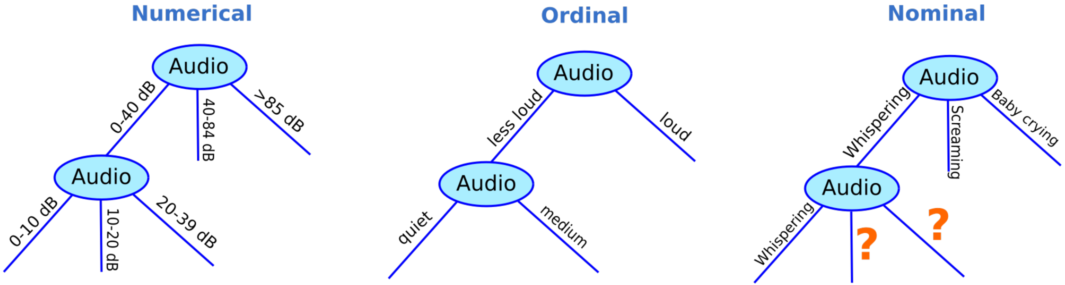
Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.



Practical issues – numeric values

Nominal feature values

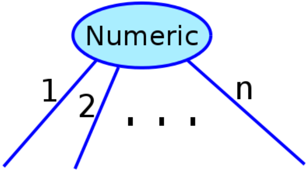
For nominal features, the decision tree splits on every possible value. Therefore, the information content of this feature is 0 after such branch has been conducted $\rightarrow$ Never branches on nominal features twice



Practical issues – numeric values

Numeric feature values

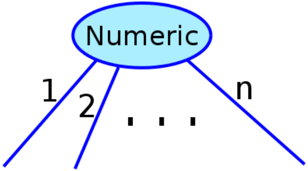
For numeric feature values, splitting on each possible value would lead to a very wide tree of small depth.



Practical issues – numeric values

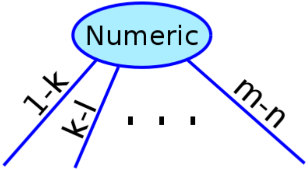
Numeric feature values

For numeric feature values, splitting on each possible value would lead to a very wide tree of small depth.



Therefore,

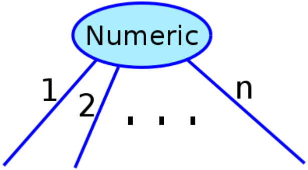
for numeric values, the tree is split into several intervals.



Practical issues – numeric values

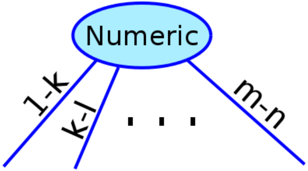
Numeric feature values

For numeric feature values, splitting on each possible value would lead to a very wide tree of small depth.

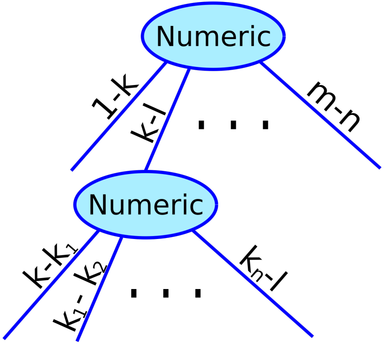


Therefore,

for numeric values, the tree is split into several intervals.



Nested intervals possible



Practical issues – Missing values

Missing values in a data set

Missing values are common in real-world data sets

- participants in a survey refuse to answer
- malfunctioning sensors
- Biology: plants or animals might die before all variables have been measured
- ...

#	WiFi	ACC	Audio	Light	Label
1	<3	walking	quiet	outdoor	Work
2	<3	walking	quiet	outdoor	Work
3	<3	walking	quiet	outdoor	Work
4	3-5	sitting	quiet	--	Work
5	3-5	sitting	quiet	indoor	Work
6	3-5	sitting	--	indoor	Work
7	3-5	sitting	quiet	indoor	Work
8	3-5	sitting	quiet	indoor	Work
9	>5	walking	loud	indoor	Work
10	>5	standing	--	indoor	Work
11	>5	sitting	medium	indoor	Work
12	>5	sitting	medium	indoor	Work
13	--	sitting	medium	indoor	Work
14	>5	sitting	medium	indoor	Work
15	>5	sitting	medium	indoor	Work
16	>5	sitting	loud	indoor	Work
17	<3	walking	--	indoor	Not at work
18	<3	walking	quiet	--	Not at work
19	<3	standing	quiet	indoor	Not at work
20	<3	walking	medium	indoor	Not at work
21	--	walking	loud	outdoor	Not at work
22	<3	walking	medium	indoor	Not at work
23	<3	walking	medium	indoor	Not at work
24	3-5	walking	quiet	outdoor	Not at work
25	3-5	standing	quiet	outdoor	Not at work
26	3-5	--	quiet	outdoor	Not at work
27	3-5	standing	loud	outdoor	Not at work
28	3-5	walking	loud	outdoor	Not at work
29	>5	sitting	loud	--	Not at work
30	--	sitting	loud	outdoor	Not at work

Practical issues – Missing values

Missing values in a data set

Missing values are common in real-world data sets

- participants in a survey refuse to answer
- malfunctioning sensors
- Biology: plants or animals might die before all variables have been measured
- ...

Most machine learning schemes assume no significance in the fact that a certain value is missing.

#	WiFi	ACC	Audio	Light	Label
1	<3	walking	quiet	outdoor	Work
2	<3	walking	quiet	outdoor	Work
3	<3	walking	quiet	outdoor	Work
4	3-5	sitting	quiet	--	Work
5	3-5	sitting	quiet	indoor	Work
6	3-5	sitting	--	indoor	Work
7	3-5	sitting	quiet	indoor	Work
8	3-5	sitting	quiet	indoor	Work
9	>5	walking	loud	indoor	Work
10	>5	standing	--	indoor	Work
11	>5	sitting	medium	indoor	Work
12	>5	sitting	medium	indoor	Work
13	--	sitting	medium	indoor	Work
14	>5	sitting	medium	indoor	Work
15	>5	sitting	medium	indoor	Work
16	>5	sitting	loud	indoor	Work
17	<3	walking	--	indoor	Not at work
18	<3	walking	quiet	--	Not at work
19	<3	standing	quiet	indoor	Not at work
20	<3	walking	medium	indoor	Not at work
21	--	walking	loud	outdoor	Not at work
22	<3	walking	medium	indoor	Not at work
23	<3	walking	medium	indoor	Not at work
24	3-5	walking	quiet	outdoor	Not at work
25	3-5	standing	quiet	outdoor	Not at work
26	3-5	--	quiet	outdoor	Not at work
27	3-5	standing	loud	outdoor	Not at work
28	3-5	walking	loud	outdoor	Not at work
29	>5	sitting	loud	--	Not at work
30	--	sitting	loud	outdoor	Not at work

Practical issues – Missing values

The absence of data might already hold valuable information!

[1] Witten et al., Data Mining, Morgan Kaufmann, 2011

Practical issues – Missing values

The absence of data might already hold valuable information!

Example

People analyzing medical databases have noticed that cases may, in some circumstances, be diagnosable simply from the tests that a doctor decides to make – regardless of the outcome of the tests[1]

[1] Witten et al., Data Mining, Morgan Kaufmann, 2011

New feature for missing values

- Add binary feature describing whether the value is missing or not
- split the instance at the missing feature:
  1. propagate all instances (weighted with the respective frequency observed from training samples) down to the leaves
  2. combine the results at the leaf nodes given the weighting of the instances

#	WiFi	ACC	Audio	Light	Label
1	<3	walking	quiet	outdoor	Work
2	<3	walking	quiet	outdoor	Work
3	<3	walking	quiet	outdoor	Work
4	3-5	sitting	quiet	--	Work
5	3-5	sitting	quiet	indoor	Work
6	3-5	sitting	--	indoor	Work
7	3-5	sitting	quiet	indoor	Work
8	3-5	sitting	quiet	indoor	Work
9	>5	walking	loud	indoor	Work
10	>5	standing	--	indoor	Work
11	>5	sitting	medium	indoor	Work
12	>5	sitting	medium	indoor	Work
13	--	sitting	medium	indoor	Work
14	>5	sitting	medium	indoor	Work
15	>5	sitting	medium	indoor	Work
16	>5	sitting	loud	indoor	Work
17	<3	walking	--	indoor	Not at work
18	<3	walking	quiet	--	Not at work
19	<3	standing	quiet	indoor	Not at work
20	<3	walking	medium	indoor	Not at work
21	--	walking	loud	outdoor	Not at work
22	<3	walking	medium	indoor	Not at work
23	<3	walking	medium	indoor	Not at work
24	3-5	walking	quiet	outdoor	Not at work
25	3-5	standing	quiet	outdoor	Not at work
26	3-5	--	quiet	outdoor	Not at work
27	3-5	standing	loud	outdoor	Not at work
28	3-5	walking	loud	outdoor	Not at work
29	>5	sitting	loud	--	Not at work
30	--	sitting	loud	outdoor	Not at work

Outline

Decision Trees

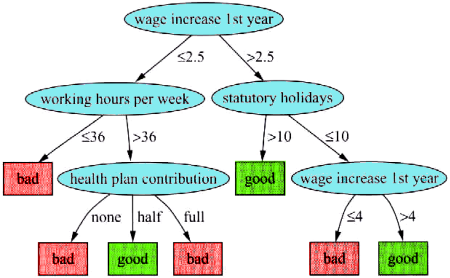
Optimizing the tree structure

Improving classification results

Optimizing the tree structure

Motivation

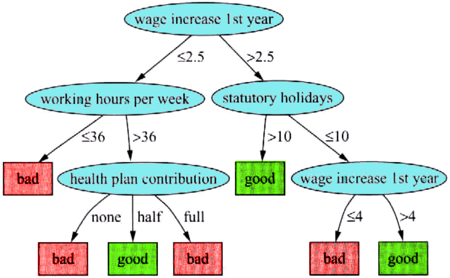
Fully expanded decision trees often contain unnecessary structure that should be simplified before deployment

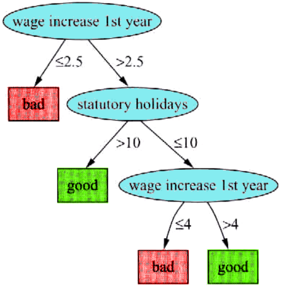


Optimizing the tree structure

Motivation

Fully expanded decision trees often contain unnecessary structure that should be simplified before deployment





Confidence on a prediction

Assume we measure the error of a classifier on a test set and estimate a numerical error rate of q' (a success rate of $p' = (1 - q')$).

What can we say about the true success rate p?

- It will be close to p',
- but how close? (within 5% or 10% ?)

This depends on the size of the test set

Naturally, we are more confident on p' when it based based on a large number of evaluations.

Confidence on a prediction

In statistics, a succession of independent events that either succeed or fail is called a Bernoulli process

Bernoulli process

A Bernoulli process is a repeated coin flipping, possibly with an unfair coin

Confidence on a prediction

In statistics, a succession of independent events that either succeed or fail is called a Bernoulli process

Assume that out of n events, s are successful.

Confidence on a prediction

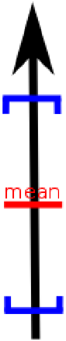
In statistics, a succession of independent events that either succeed or fail is called a Bernoulli process

Assume that out of n events, s are successful.

Then we have an observed success rate of $p' = \frac{s}{n}$

Confidence on a prediction

In statistics, a succession of independent events that either succeed or fail is called a Bernoulli process



Assume that out of n events, s are successful.

Then we have an observed success rate of $p' = \frac{s}{n}$

Confidence Interval

The true success rate p lies within an interval with a specified confidence

Confidence on a prediction

(σ and μ are the standard deviation and mean of p')

The probability that a random variable $\overline{p} = \frac{p' - \mu}{\sigma}$, with zero mean and unit variance, lies within a certain confidence range of width $2z$ is

$$\mathcal{P}[-z \leq \overline{p} \leq z] = c$$

Confidence on a prediction

(σ and μ are the standard deviation and mean of p')

The probability that a random variable $\overline{p} = \frac{p' - \mu}{\sigma}$, with zero mean and unit variance, lies within a certain confidence range of width $2z$ is

$$\mathcal{P}[-z \leq \overline{p} \leq z] = c$$

Confidence limits for the normal distribution are e.g.

$\mathcal{P}[\overline{p} \geq z]$	0.001	0.005	0.01	0.05	0.1	0.2	0.4
z	3.09	2.58	2.33	1.65	1.28	0.84	0.25

Standard assumption in such tables on the random variable:

mean 0
variance 1

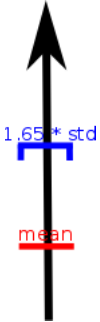
Confidence on a prediction

$\mathcal{P}[\overline{p} \geq z]$	0.001	0.005	0.01	0.05	0.1	0.2	0.4
z	3.09	2.58	2.33	1.65	1.28	0.84	0.25

z is measured in standard deviations from the mean:

Confidence on a prediction

$\mathcal{P}[\overline{p} \geq z]$	0.001	0.005	0.01	0.05	0.1	0.2	0.4
z	3.09	2.58	2.33	1.65	1.28	0.84	0.25



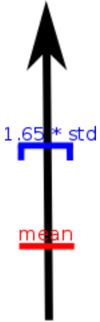
z is measured in standard deviations from the mean:

Interpretation

E.g. $\mathcal{P}[\overline{p} \geq z] = 0.05$ implies that there is a 5% chance that $\overline{p}$ lies more than 1.65 standard deviations above the mean.

Confidence on a prediction

$\mathcal{P}[\overline{p} \geq z]$	0.001	0.005	0.01	0.05	0.1	0.2	0.4
z	3.09	2.58	2.33	1.65	1.28	0.84	0.25



z is measured in standard deviations from the mean:

Interpretation

E.g. $\mathcal{P}[\overline{p} \geq z] = 0.05$ implies that there is a 5% chance that $\overline{p}$ lies more than 1.65 standard deviations above the mean.

Since the distribution is symmetric, the chance that $\overline{p}$ lies more than 1.65 standard deviations from the mean is 10%:

$$\mathcal{P}[-1.65 \leq \overline{p} \leq 1.65] = 0.9$$

Confidence on a prediction

In order to apply this to the random variable p', we have to reduce it to have zero mean and unit variance.

Confidence on a prediction

In order to apply this to the random variable p', we have to reduce it to have zero mean and unit variance.

$\rightarrow$ subtract mean μ & divide by standard deviation $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(p' - \mu)^2}{n}}$

Confidence on a prediction

In order to apply this to the random variable p', we have to reduce it to have zero mean and unit variance.

$\rightarrow$ subtract mean μ & divide by standard deviation $\sigma = \sqrt{\frac{\sum_{i=1}^{n}(p'-\mu)^2}{n}}$

This leads to

$$\mathcal{P}\left[-z < \frac{p' - \mu}{\sqrt{\frac{\sum_{i=1}^{n}(p'-\mu)^2}{n}}} < z\right] = c$$

Confidence on a prediction

To find confidence limits z, given a target confidence value c:

- consult a table with confidence limits for the normal distribution

Table 5.1 Confidence Limits for the Normal Distribution

$Pr[X \geq z]$	z
0.1%	3.09
0.5%	2.58
1%	2.33
5%	1.65
10%	1.28
20%	0.84
40%	0.25

Confidence on a prediction

To find confidence limits z, given a target confidence value c:

- consult a table with confidence limits for the normal distribution
- since one-sided *success* probabilities (not *error*-) are displayed, we have to subtract $Pr[X \geq z] = c$ from 1 and divide by two:

$$z = \frac{1 - c}{2}$$

Confidence on a prediction

$$\mathcal{P}\left[-z < \frac{p' - \mu}{\sqrt{\frac{\sum_{i=1}^{n}(p'-\mu)^2}{n}}} < z\right] = c$$

- Then, write inequality above as equality, invert it to find an expression for μ and solve a quadratic equation to yield

$$\mu = \frac{\left(p' + \frac{z^2}{2n} \pm z\sqrt{\frac{p'}{n} - \frac{p'^2}{n} + \frac{z^2}{4n^2}}\right)}{1 + \frac{z^2}{n}}$$

The resulting two values are the upper and lower confidence boundaries

Confidence on a prediction

Example

$$p' = 0.75;\; n = 1000,\; c = 0.8\;(z = 1.28) \;\rightarrow\; [0.732, 0.767]$$
$$p' = 0.75;\; n = 100,\; c = 0.8\;(z = 1.28) \;\rightarrow\; [0.691, 0.801]$$

Note that the assumptions taken are only valid for large n

Optimization – Noisy data

Fully expanded decision trees often contain unnecessary structure that should be simplified before deployment

Optimization – Noisy data

Fully expanded decision trees often contain unnecessary structure that should be simplified before deployment

Pruning

Prepruning Trying to decide through the tree-building process when to stop developing subtrees

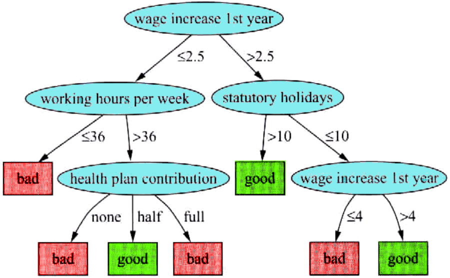
- Might speed up tree creation phase
- Difficult to spot dependencies between features at this stage (features might be meaningful together but not on their own)

Postpruning Simplification of the decision tree after the tree has been created

Postpruning – subtree replacement

Select some subtrees and replace them with single leaves

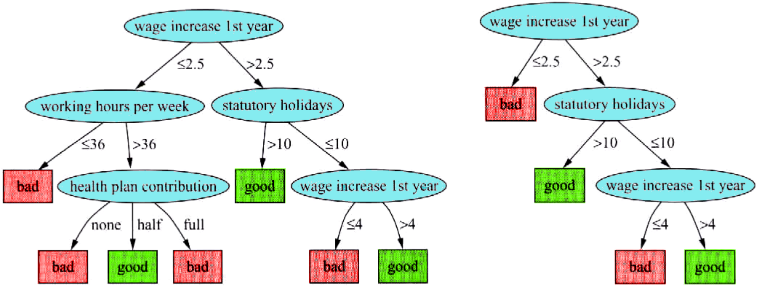
- Will reduce accuracy on the training set
- May increase accuracy on independently chosen test set (reduction of noise)



Postpruning – subtree replacement

Select some subtrees and replace them with single leaves

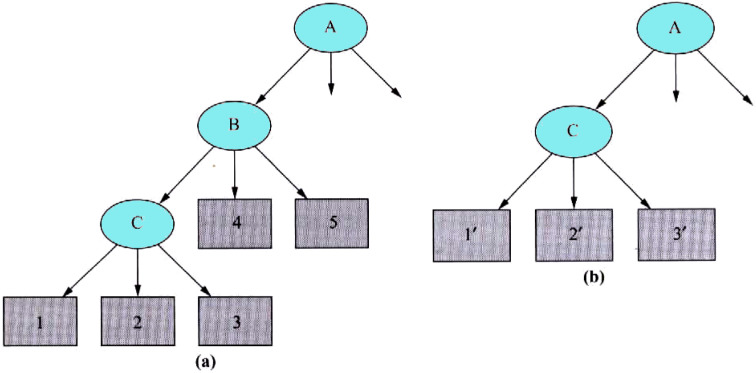
- Will reduce accuracy on the training set
- May increase accuracy on independently chosen test set (reduction of noise)



Optimization – Noisy data

Postpruning – subtree raising

Complete subtree is raised one level and samples at the nodes of the subtree have to be recalculated



Optimization – Estimating error rates

When should we raise or replace subtrees?

Optimization – Estimating error rates

When should we raise or replace subtrees?

Estimating error rates

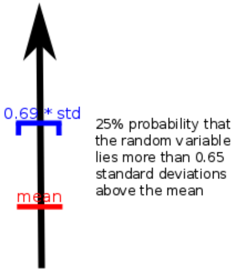
Raise the tree, when the estimated error rate of an expanded tree (considering all leaf nodes) would exceed the estimated error rate of a raised subtree.

Estimating error rates – success probability

Given a confidence c we find a confidence limit z (for $c = 25\% \rightarrow z = 0.69$) such that

$$\mathcal{P}\left[\frac{q' - \mu_{q'}}{\sqrt{\frac{q'(1-q')}{n}}} > z\right] = c$$

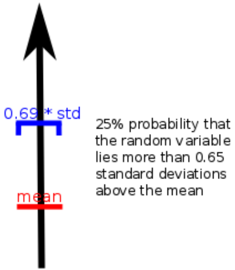
(with the observed error rate $q' = \frac{e}{n}$)



Estimating error rates – success probability

Given a confidence c we find a confidence limit z (for $c = 25\% \rightarrow z = 0.69$) such that

$$\mathcal{P}\left[\frac{q' - \mu_{q'}}{\sqrt{\frac{q'(1-q')}{n}}} > z\right] = c$$



(with the observed error rate $q' = \frac{e}{n}$)

- This leads to a pessimistic error rate $\mu_{q'}$ as an upper confidence limit for q (solving the equation for q):

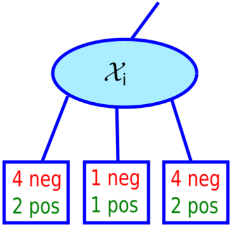
$$\mu_{q'} = \frac{q' + \frac{z^2}{2n} + z\sqrt{\frac{q'}{n} - \frac{q'^2}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}}$$

Example

Lower left leaf ($e = 2, n = 6$) Utilising the formula for $\mu_{q'}$, we obtain
$q' = 0.33$ and $\mu_{q'} = 0.47$

Minimizing the error:

Majority vote at the parent node F1 vs. majority votes at the leaves ?



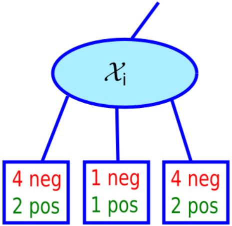
Example

Lower left leaf ($e = 2, n = 6$) Utilising the formula for $\mu_{q'}$, we obtain
$q' = 0.33$ and $\mu_{q'} = 0.47$

Center leaf($e = 1, n = 2$) $\mu_{q'} = 0.72$

Minimizing the error:

Majority vote at the parent node F1 vs. majority votes at the leaves ?



Example

Lower left leaf ($e = 2, n = 6$) Utilising the formula for $\mu_{q'}$, we obtain

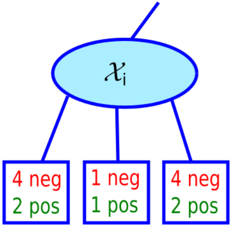
$q' = 0.33$ and $\mu_{q'} = 0.47$

Center leaf($e = 1, n = 2$) $\mu_{q'} = 0.72$

Right leaf ($e = 2, n = 6$) $\mu_{q'} = 0.47$

Minimizing the error:

Majority vote at the parent node F1 vs. majority votes at the leaves ?



Example

Lower left leaf ($e = 2, n = 6$) Utilising the formula for $\mu_{q'}$, we obtain $q' = 0.33$ and $\mu_{q'} = 0.47$

Center leaf($e = 1, n = 2$) $\mu_{q'} = 0.72$

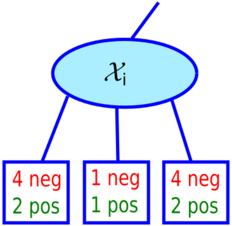
Right leaf ($e = 2, n = 6$) $\mu_{q'} = 0.47$

Combine error estimates Utilising ratio 6:2:6 this leads to a combined error estimate of

$$\frac{0.47 \cdot 6}{14} + \frac{0.72 \cdot 2}{14} + \frac{0.47 \cdot 6}{14} \approx 0.51$$

Minimizing the error:

Majority vote at the parent node F1 vs. majority votes at the leaves ?



Example

Lower left leaf ($e = 2, n = 6$) Utilising the formula for $\mu_{q'}$, we obtain $q' = 0.33$ and $\mu_{q'} = 0.47$

Center leaf($e = 1, n = 2$) $\mu_{q'} = 0.72$

Right leaf ($e = 2, n = 6$) $\mu_{q'} = 0.47$

Combine error estimates Utilising ratio 6:2:6 this leads to a combined error estimate of

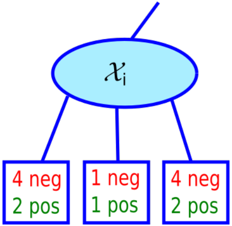
$$\frac{0.47 \cdot 6}{14} + \frac{0.72 \cdot 2}{14} + \frac{0.47 \cdot 6}{14} \approx 0.51$$

Error estimate for parent node $q' = \frac{5}{14} \to \mu_{q'} = 0.46$

$0.46 < 0.51 \Rightarrow$ prune children away

Minimizing the error:

Majority vote at the parent node F1 vs. majority votes at the leaves ?



Outline

Decision Trees

Optimizing the tree structure

Improving classification results

Bottom-line: Decision trees

Strengths

- Simple, intuitive approach
- Robust to the inclusion of irrelevant features
- Invariant under transformation of features, e.g. scaling

Bottom-line: Decision trees

Strengths

- Simple, intuitive approach
- Robust to the inclusion of irrelevant features
- Invariant under transformation of features, e.g. scaling

Weaknesses

- Tendency to overfit
- Often complex, deep trees even for simple linearly separable classes

Improving classification results

C4.5 – design decisions ($\rightarrow$ heuristic)

Postpruning – Confidence value $c = 25\%$

Postpruning – Split Threshold Candidate splits on a numeric feature are only considered when at least $\min(10\%, 25)$ of all training samples are cut off by the split

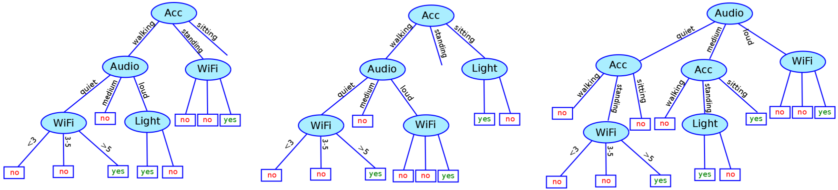
Prepruning with information gain Given u candidate splits on a certain numeric attribute, $\log_2 \frac{u}{n}$ is subtracted from the information gain

- in order to prevent overfitting
- Negative information gain $\rightarrow$ tree-construction will stop

Improving classification results

Tree bagging

Bootstrap aggregating, or bagging builds several 100 or 1000 trees from random subsets of the training set (*random samples with replacement*)



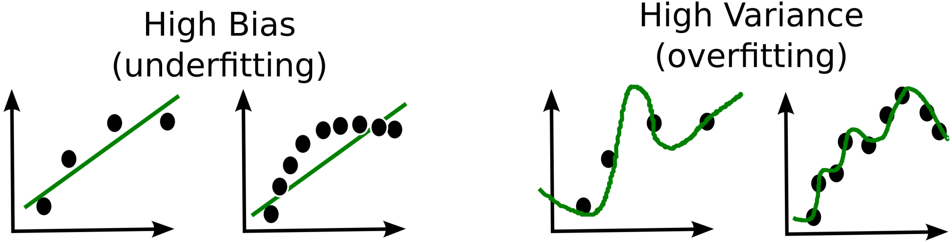
Predictions are made after majority vote or by averaging probabilities.

Reduces variance without affecting bias

Improving classification results

Tree bagging

Bootstrap aggregating, or bagging builds several 100 or 1000 trees from random subsets of the training set (*random samples with replacement*)



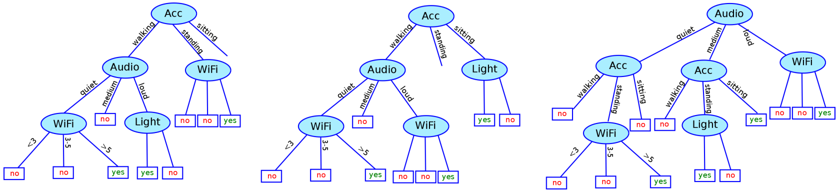
Predictions are made after majority vote or by averaging probabilities.

Reduces variance without affecting bias

Improving classification results

Random forests

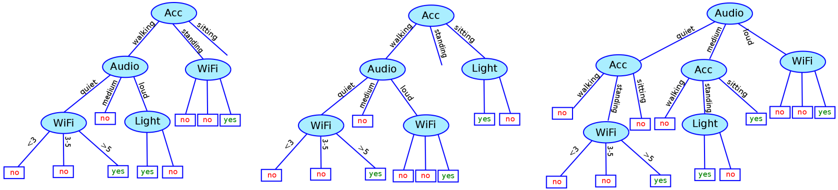
Random forests exploit *Tree bagging* and in addition use a random subset of features at each candidate split in order to reduce the impact of strong features. (Strong features may lead to dependent trees and thus impair the benefits of Tree bagging)



Improving classification results

Extra Trees

A way to generate extremely randomized trees is to build a *Random forest* but in addition for each feature split exploit random decision (based on *information gain* or *Gini impurity*) instead of deterministic choice.



Questions?

Stephan Sigg
stephan.sigg@aalto.fi

Si Zuo
si.zuo@aalto.fi

Literature

I.H. Witten, E. Frank, M.A. Hall: Data Mining – Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2011.