

Analysis of Microarray Data with Methods from Machine Learning and Network Theory

Summer Lecture 2015

Prof. Dr. A. B. Cremers

Dr. Jörg Zimmermann

Supervised Learning: Decision Trees

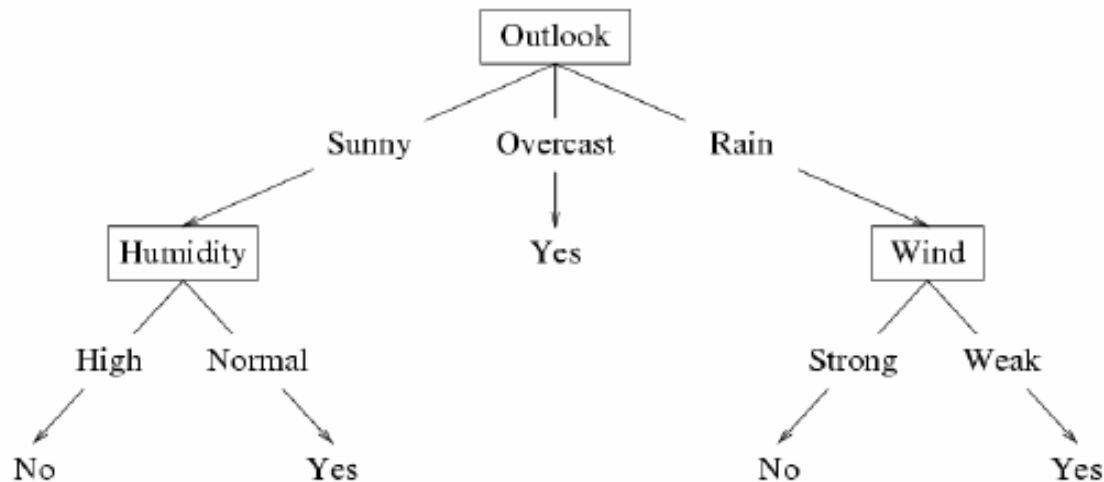
1. Decision Trees are a supervised learning method and are used to **predict** known class labels for new data points
2. In the following, we will deal with:
 - * Decision Tree Representation
 - * Attribute Selection, Entropy, Information Gain
 - * Overfitting

Training Data Example: When Does Greg Play Tennis?

Day	Outlook	Temp.	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

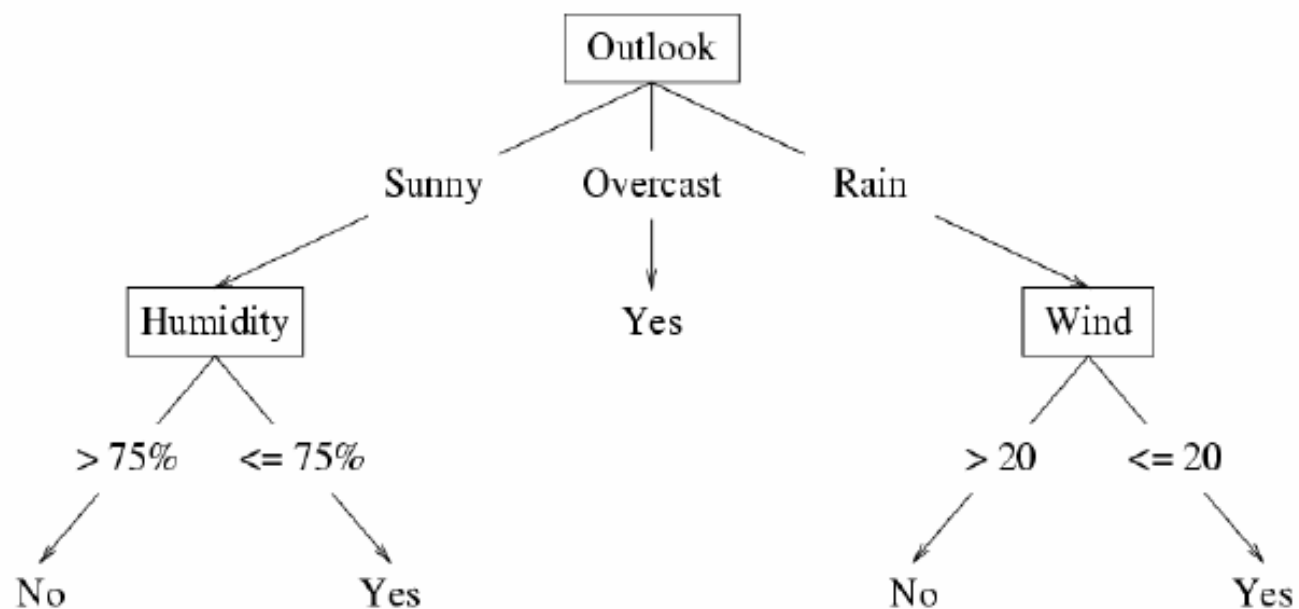
Decision Tree Hypothesis Space

- **Internal nodes** test the value of particular features x_j and branch according to the results of the test.
- **Leaf nodes** specify the class $h(\mathbf{x})$.



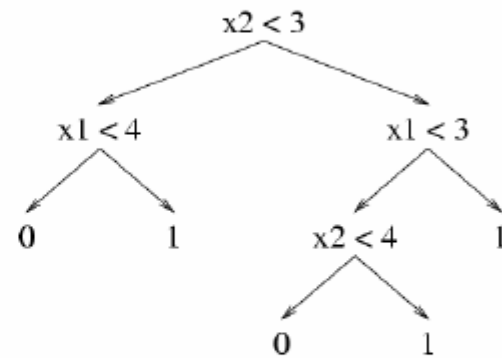
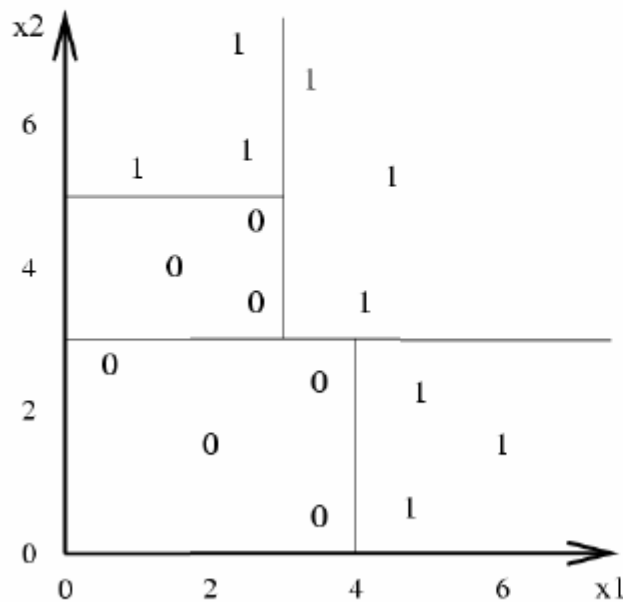
Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (\text{Sunny}, \text{Hot}, \text{High}, \text{Strong})$ will be classified as **No**. The **Temperature** feature is irrelevant.

If the features are continuous, internal nodes may test the value of a feature against a threshold.

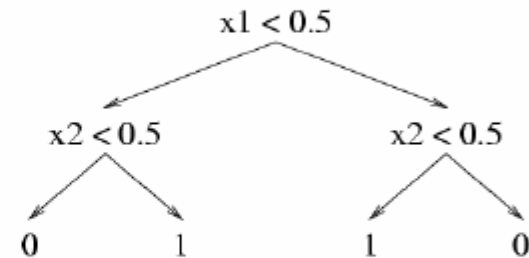
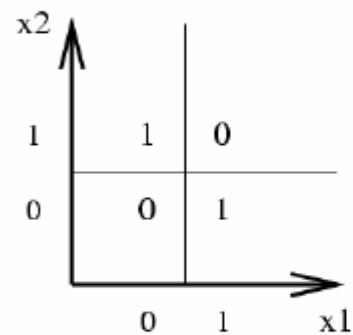


Decision Tree Decision Boundaries

Decision trees divide the feature space into axis-parallel rectangles, and label each rectangle with one of the K classes.



Decision Trees Can Represent Any Boolean Function



The tree will in the worst case require exponentially many nodes, however.

Learning Algorithm for Decision Trees

$$S = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\} \quad \begin{array}{l} \mathbf{x} = (x_1, \dots, x_d) \\ x_j, y \in \{0, 1\} \end{array}$$

GROWTREE(S)

if ($y = 0$ for all $\langle \mathbf{x}, y \rangle \in S$) **return** new leaf(0)

else if ($y = 1$ for all $\langle \mathbf{x}, y \rangle \in S$) **return** new leaf(1)

else

 choose best attribute x_j

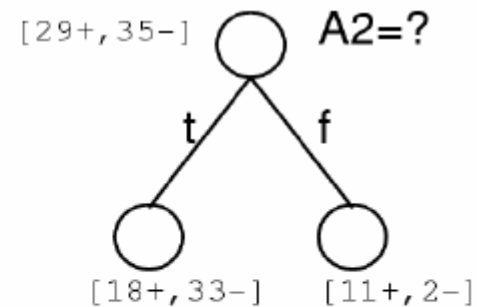
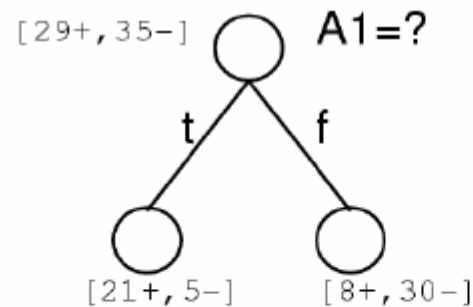
$S_0 =$ all $\langle \mathbf{x}, y \rangle \in S$ with $x_j = 0$;

$S_1 =$ all $\langle \mathbf{x}, y \rangle \in S$ with $x_j = 1$;

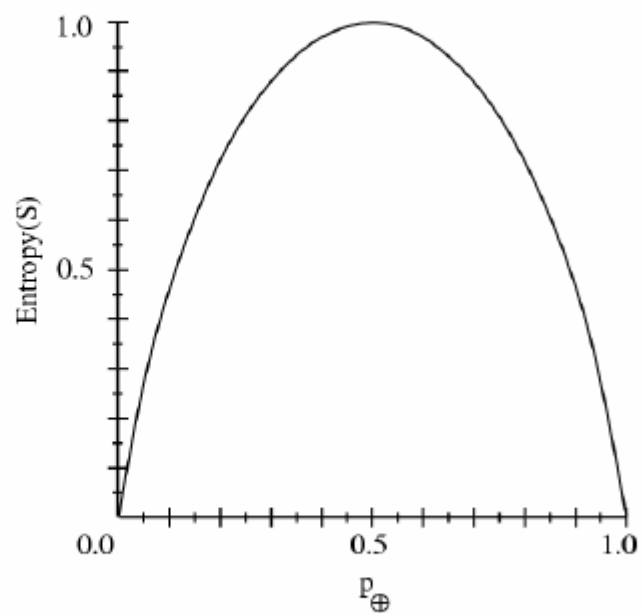
return new node(x_j , GROWTREE(S_0), GROWTREE(S_1))

Choosing the **Best** Attribute

Which attribute is best?



Entropy



- S is a sample of training examples

Entropy

- p_{\oplus} is the proportion of positive examples in S
- p_{\ominus} is the proportion of negative examples in S
- Entropy measures the impurity of S

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Entropy

$Entropy(S)$ = expected number of bits needed to encode class (\oplus or \ominus) of randomly drawn member of S (under the optimal, shortest-length code)

Why?

Information theory: optimal length code assigns $-\log_2 p$ bits to message having probability p .

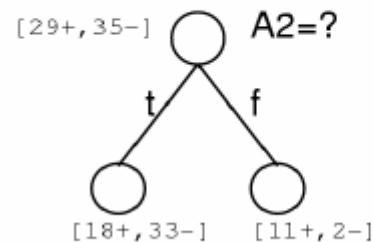
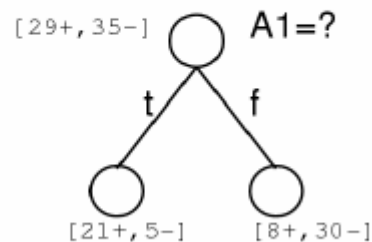
So, expected number of bits to encode \oplus or \ominus of random member of S :

$$p_{\oplus}(-\log_2 p_{\oplus}) + p_{\ominus}(-\log_2 p_{\ominus})$$
$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Information Gain

$Gain(S, A)$ = expected reduction in entropy due to sorting on A

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

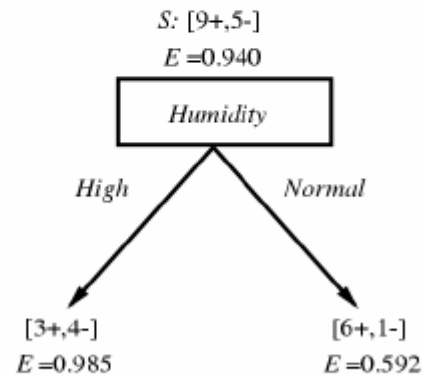


Training Example

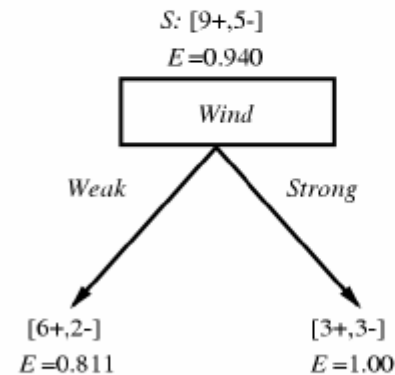
Day	Outlook	Temp.	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Selecting the Next Attribute

Which attribute is the best classifier?



$$\begin{aligned} \text{Gain}(S, \text{Humidity}) &= .940 - (7/14) \cdot .985 - (7/14) \cdot .592 \\ &= .151 \end{aligned}$$



$$\begin{aligned} \text{Gain}(S, \text{Wind}) &= .940 - (8/14) \cdot .811 - (6/14) \cdot 1.0 \\ &= .048 \end{aligned}$$

Non-Boolean Features

- Features with multiple discrete values
 - Multi-way splits
 - Test for one value versus the rest
 - Group values into disjoint sets
- Real-valued features
 - Use thresholds
- Regression
 - Splits based on mean squared error metric

Overfitting

Consider error of hypothesis h over

- training data: $error_{train}(h)$
- entire distribution \mathcal{D} of data: $error_{\mathcal{D}}(h)$

Hypothesis $h \in H$ **overfits** training data if there is an alternative hypothesis $h' \in H$ such that

$$error_{train}(h) < error_{train}(h')$$

and

$$error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h')$$

Overfitting in Decision Trees

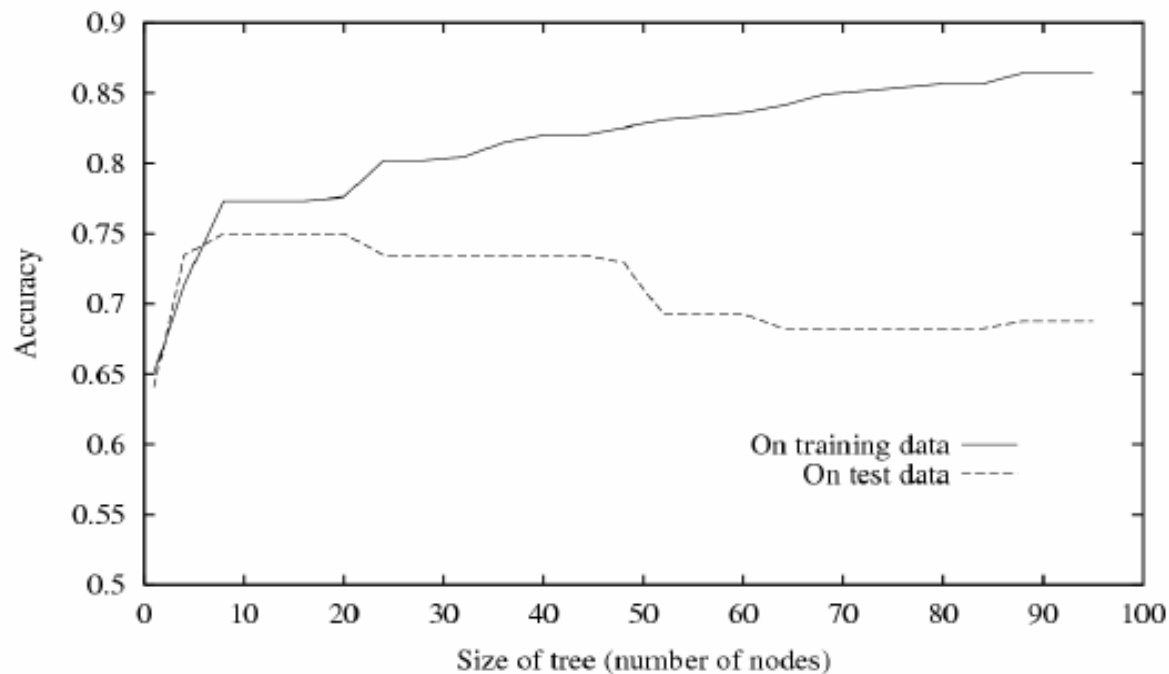
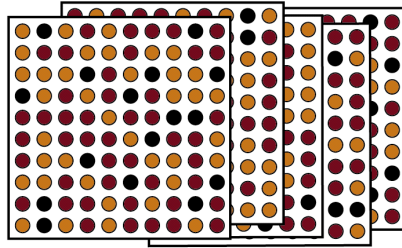


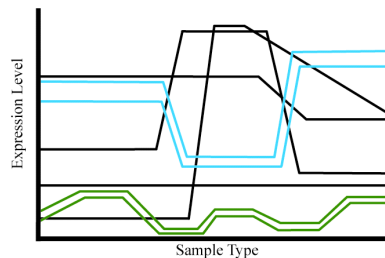
Figure 1

A Array Data



Data contains correlations

B Correlation Analysis



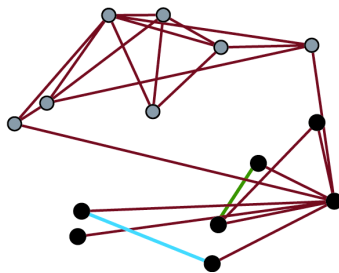
Correlation coefficients for all genes

C Correlation Matrix

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
G1	1	0.9	0.9	0.9	0.9	0.8	0.9	0.1	0.9	0.1	0.1	0.8	0.2	0.2
G2	0.9	1	0.9	0.3	0.3	0.7	0.0	0.5	0.3	0.1	0.1	0.2	0.4	0.3
G3	0.9	0.9	1	0.9	0.0	0.2	0.5	0.7	0.6	0.5	0.2	0.6	0.1	0.0
G4	0.9	0.3	0.9	1	0.5	0.3	0.6	0.3	0.0	0.5	0.1	0.2	0.2	0.6
G5	0.9	0.3	0.0	0.5	1	0.1	0.6	0.1	0.3	0.3	0.3	0.5	0.2	0.5
G6	0.8	0.7	0.2	0.3	0.1	1	0.9	0.2	0.1	0.1	0.5	0.3	0.1	0.1
G7	0.9	0.0	0.5	0.6	0.6	0.9	1	0.3	0.1	0.5	0.1	0.3	0.5	0.2
G8	0.1	0.5	0.7	0.3	0.1	0.2	0.3	1	0.9	0.9	0.9	0.8	0.8	0.9
G9	0.9	0.3	0.6	0.0	0.3	0.1	0.1	0.9	1	0.8	0.1	0.3	0.5	0.3
G10	0.1	0.1	0.5	0.5	0.3	0.1	0.5	0.9	0.8	1	0.8	1.0	0.2	0.3
G11	0.1	0.1	0.2	0.1	0.3	0.5	0.1	0.9	0.1	0.8	1	0.5	0.8	0.9
G12	0.8	0.2	0.6	0.2	0.5	0.3	0.3	0.8	0.3	1.0	0.5	1	0.8	0.1
G13	0.2	0.4	0.1	0.2	0.2	0.1	0.5	0.8	0.5	0.2	0.8	0.8	1	0.9
G14	0.2	0.3	0.0	0.6	0.5	0.1	0.2	0.9	0.3	0.3	0.9	0.1	0.9	1

Convert into Adjacency Matrix and Network

D Coexpression Network



Network Inference: Steps for constructing a correlation network

- Gene expression data
- Measure concordance of gene expression with a Pearson correlation
- The Pearson correlation matrix is either dichotomized to arrive at an unweighted adjacency matrix → unweighted network
Or transformed continuously with the power adjacency function → weighted network