Machine Learning

Prof. Dr. Stefan Kramer
Johannes Gutenberg-Universität Mainz

Introduction to Machine Learning

Prof. Dr. Stefan Kramer
Johannes Gutenberg-Universität Mainz

Organization

- Machine Learning (2+2)
- **Time:** Thu 14:15-16:00
- **Place:** 03-428
- Tutorial by:
  Z. Ahmadi, J. Vexler, L. Nathan:
  Wed 16:15-17:45 and Mon/Fri (cf. doodle, tbd)
- Staudingerweg 9, 3. Stock, North wing
- ***Course materials and submission of tutorial solutions (see next slide) in moodle.uni-mainz.de***

Tutorial

- Exercise sheets: available on Thursdays and are usually due on the subsequent Thursday. **Please submit your solutions via Moodle and put a copy of them into the submission box (Abgabekasten).**
- **You may and should submit in groups of three.**

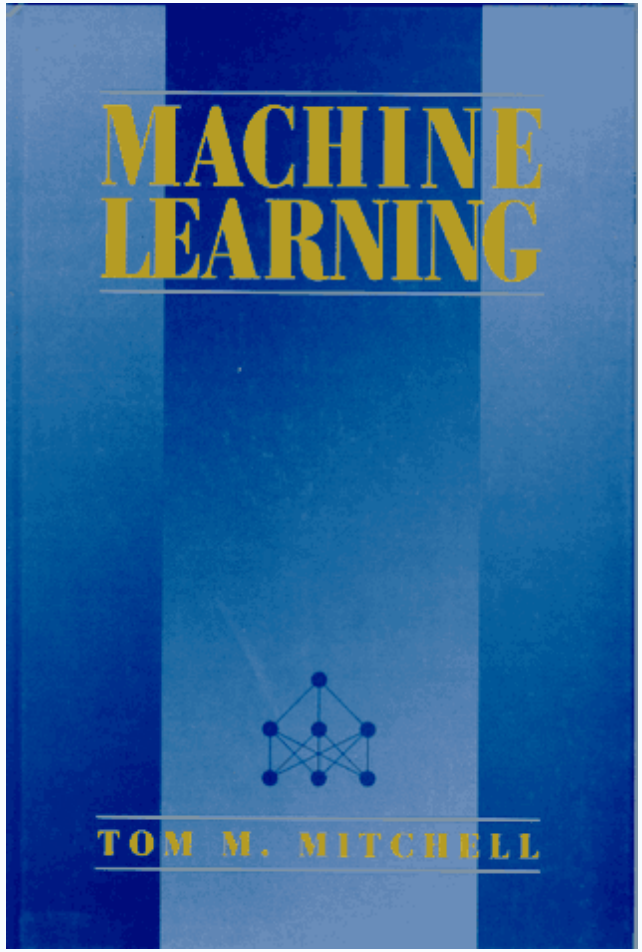
Tutorial

- *Theory*
- *Application* of machine learning and data mining tools (Weka, …) to real-world data.
- Most likely several ‚free style' programming projects (solve problem x)

Exam

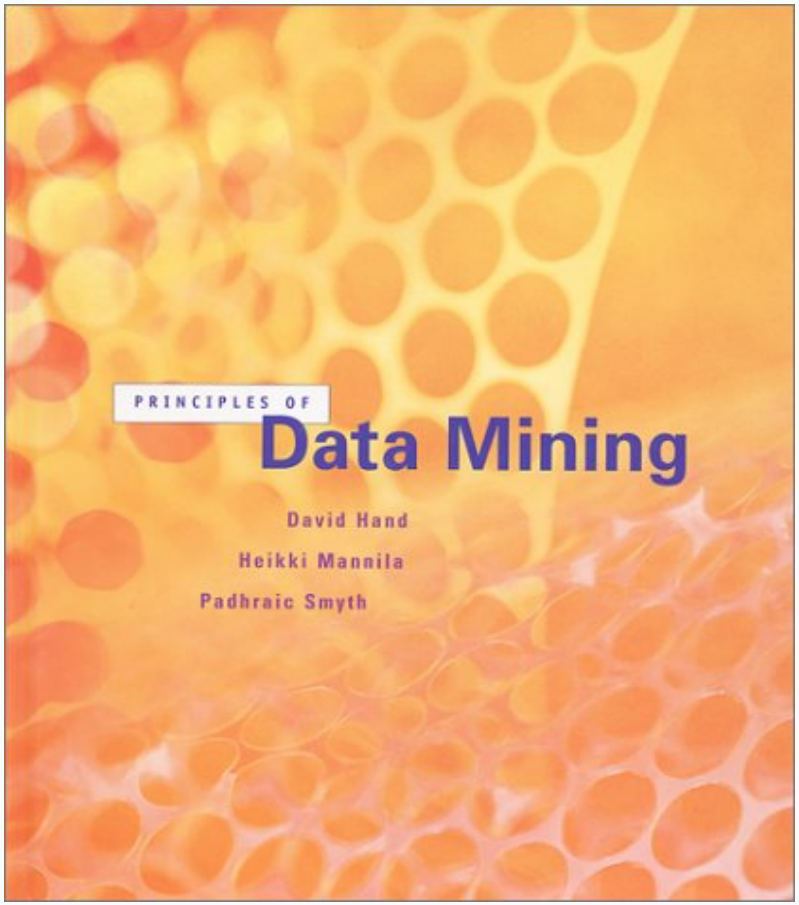
- Written exam (Klausur) after teaching term
- Admission to the exam is granted, if at least 25% of the points is achieved on each exercise sheet and overall at least 50% of the points.

Machine Learning, Tom Mitchell, McGraw Hill, 1997

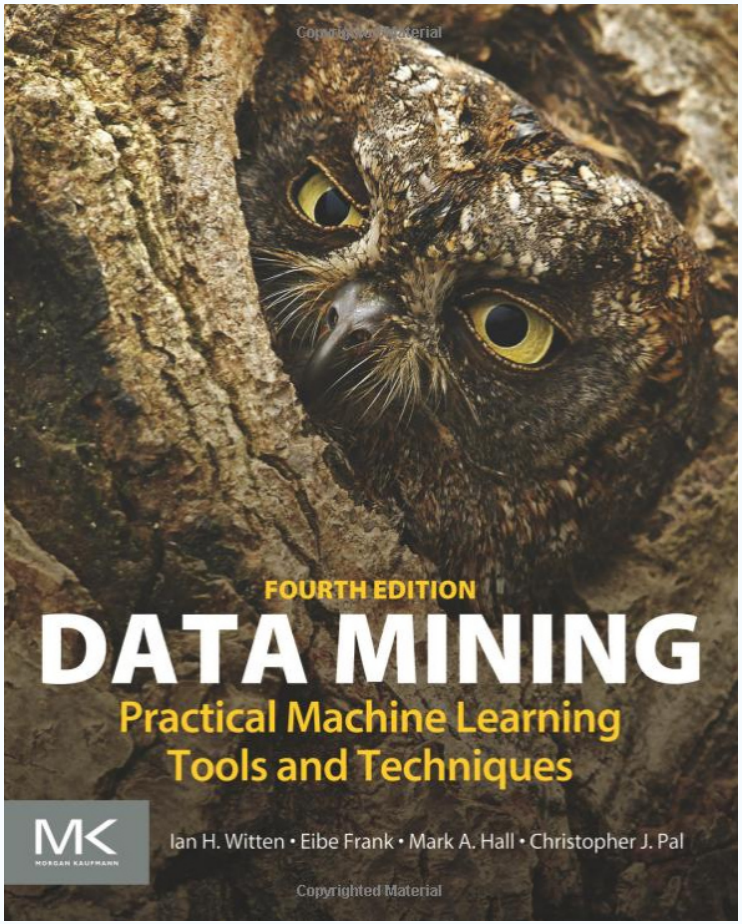


1. Introduction
2. Concept Learning and the General-to-Specific Ordering
3. Decision Tree Learning
4. Artificial Neural Networks
5. Evaluating Hypotheses
6. Bayesian Learning
7. Computational Learning Theory
8. Instance-Based Learning
9. Genetic Algorithms
10. Learning Sets of Rules
11. Analytical Learning
12. Combining Inductive and Analytical Learning
13. Reinforcement Learning

Principles of Data Mining, David Hand, Heikki Mannila, Padhraic Smyth, MIT Press, 2001

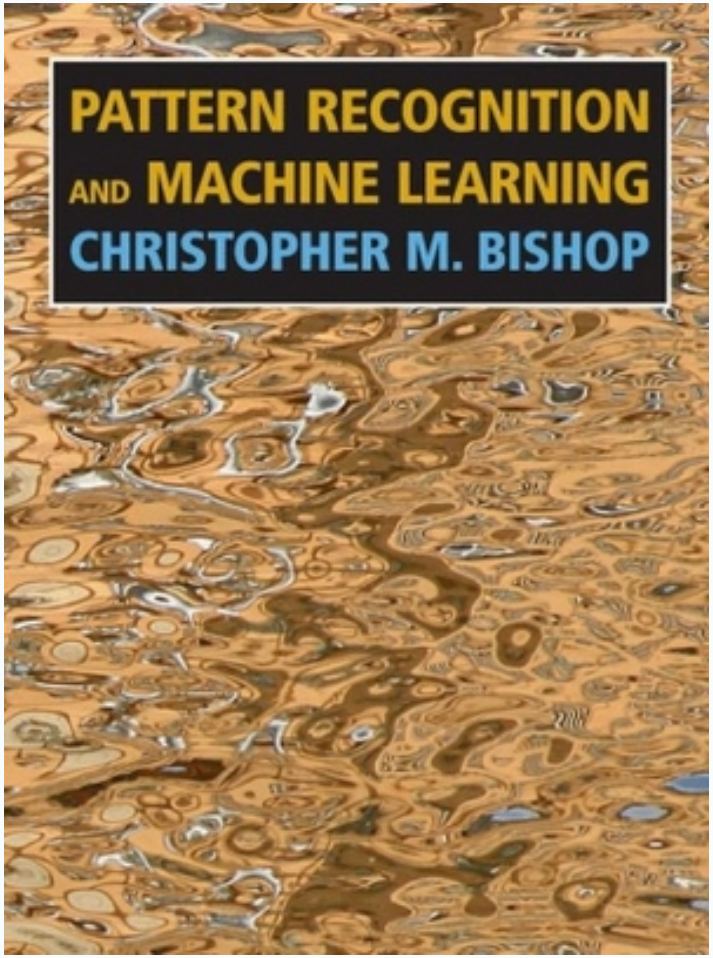


Data Mining: Practical Machine Learning Tools and Techniques with Java Implementions,
Ian H. Witten, Eibe Frank, Mark Hall, Christopher Pal
Morgan Kaufmann, 2016 (4th Edition)



Parts on validation, clustering, linear models and SVMs. Also relevant for exercises (WEKA workbench).

Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 2006



Machine Learning

- Learning = improving with experience at some task
  - Improve on task T
  - With respect to performance measure P
  - Based on experience E.
- Learn to play checkers (Dame):
  - T: Play checkers
  - P: % of games won
  - E: opportunity to play against oneself

Machine Learning

- Learning to classify examples (e.g., gene expression profiles into two subtypes):
  - T: Classifying examples
  - P: % of examples classified correctly
  - E: Training set of examples to learn from
- Machine learning algorithms (such as for classification) often used in Data Mining

Machine Learning Course Overview

- decision trees: representation, learning, overfitting, pruning
- ensembles: boosting, bagging, stacking, random forests
- evaluation and validation (abridged)
- Bayesian learning: Bayes optimality, MDL, Naive Bayes, brief introduction to Bayesian networks and learning of BNs
- linear models: linear regression, ridge regression, logistic regression
- neural networks: perceptron, multi-layer perceptron, back propagation
- instance-based learning: k-NN, locally weighted learning, RBF networks, case-based reasoning
- SVMs: margins, kernels

KW	Wed / ?	Thu
16 (16.04. - 22.04.)		VO
17 (23.04. – 29.04.)	-	VO
18 (30.04. – 06.05.)	UE	VO
19 (07.05. – 13.05.)	UE	Christi Himmelfahrt
20 (14.05. – 20.05.)	UE	VO
21 (21.05. – 27.05.)	UE	VO
22 (28.05. – 03.06.)	UE	Fronleichnam
23 (04.06. – 10.06.)	UE	VO
24 (11.06. – 17.06.)	UE	VO
25 (18.06. – 24.06.)	UE	VO
26 (25.06. – 01.07.)	UE	VO
27 (02.07. – 08.07.)	UE	VO

Outline

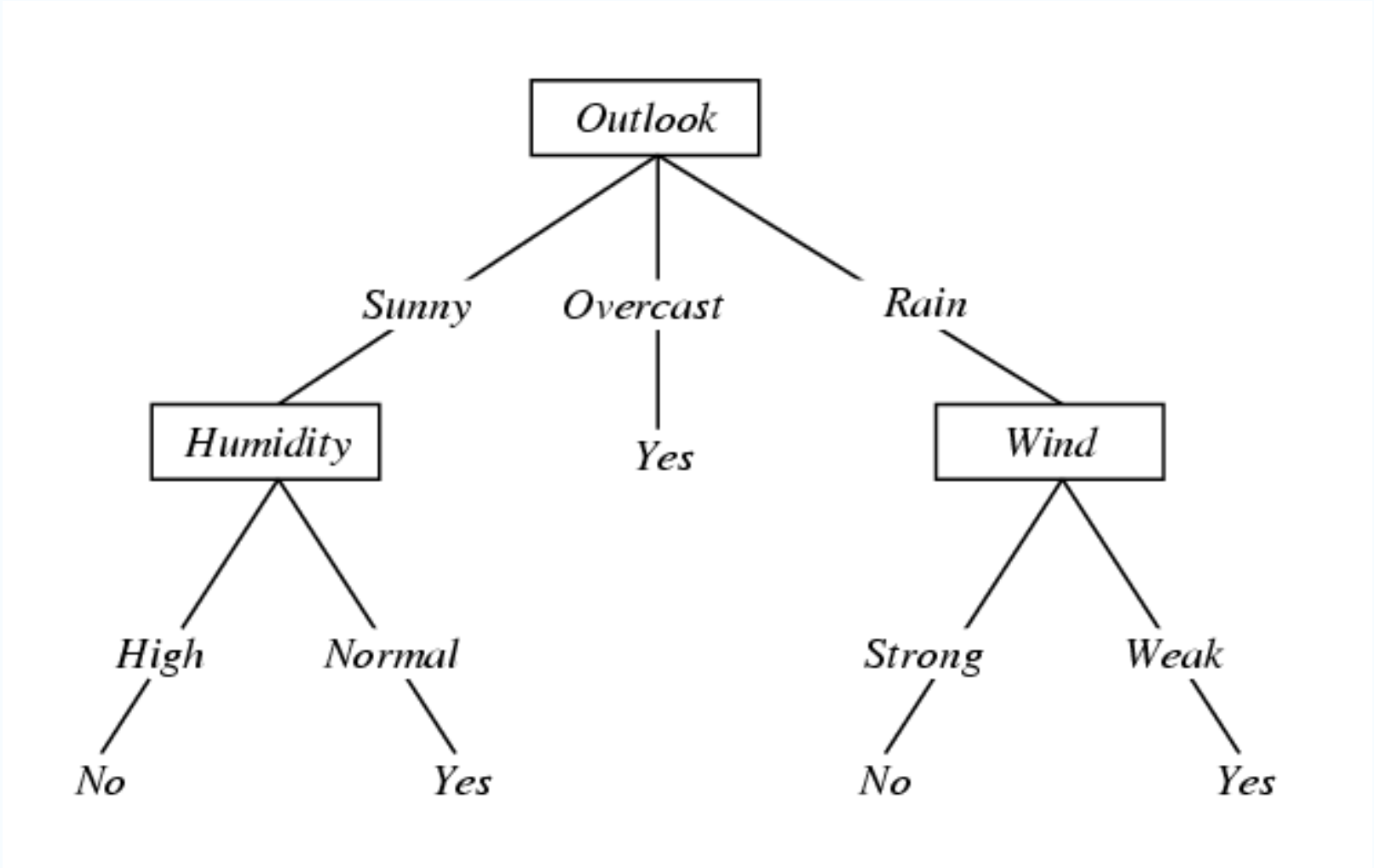
- Decision tree learning
- Choosing attributes: entropy and information gain
- Overfitting avoidance and decision tree pruning
- Other issues in decision tree learning

Decision Tree Learning

Example Dataset

Day	Outlook	Temp.	Hum.	Wind	PlayT.
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

Example Tree



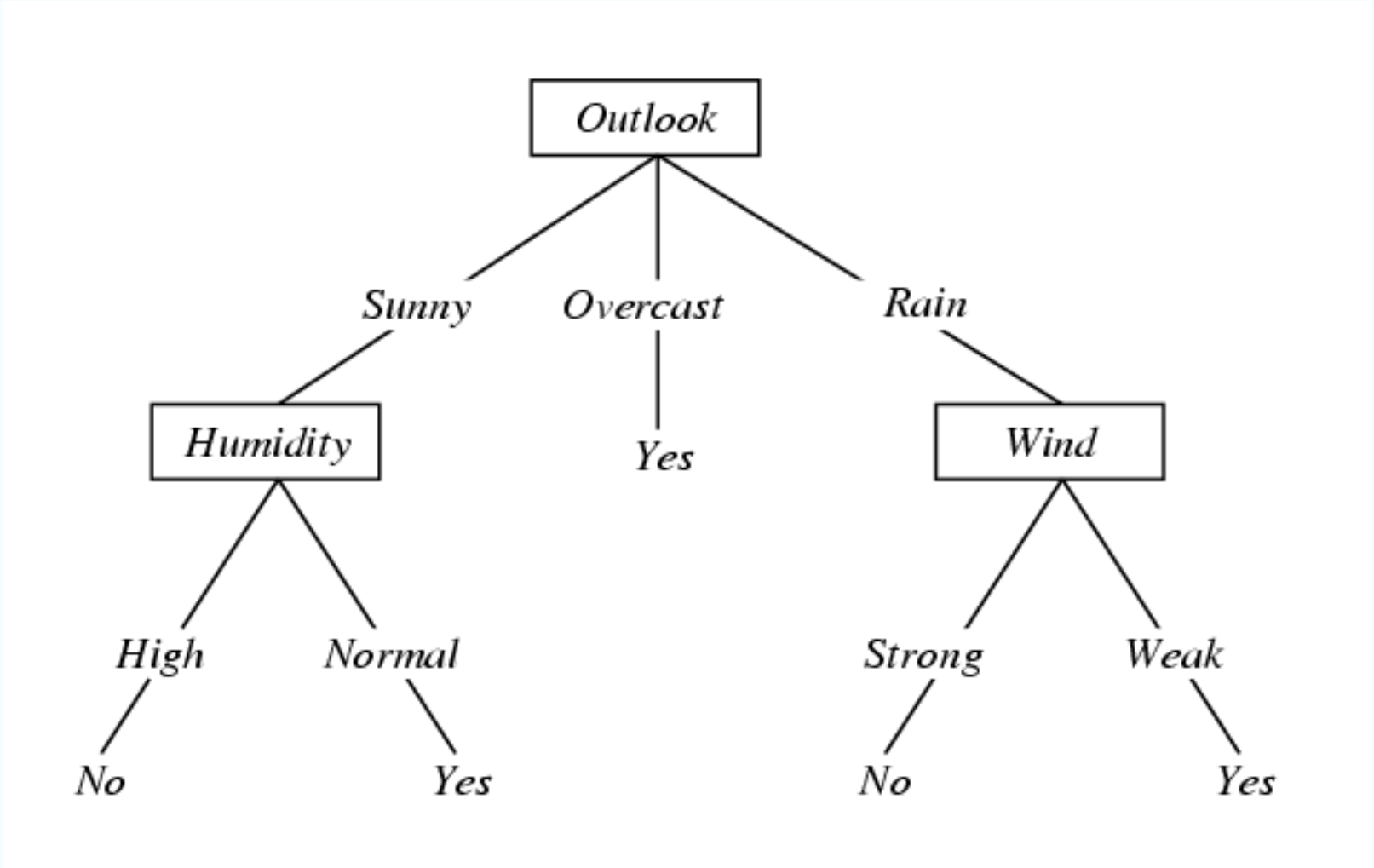
Example Dataset

Day	Outlook	Temp.	Hum.	Wind	PlayT.
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

Training Set

Test Set

Example Tree



Example Tree to Predict C-Section Risk

[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .05-
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+ .22-
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-

Decision Tree Representation

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

How would we represent?

- and, or, XOR
- (A and B) or (C and not D and E)
- M of N

When to Consider Decision Trees

- Instances describable by attribute-value pairs
- Target function is *discrete* valued
- *Disjunctive* hypothesis may be required
- Possibly *noisy* training data

Examples:

- Medical diagnosis
- Credit risk analysis
- …

When to Consider Decision Trees As Well

To every rule, there is an exception…

- "Instances describable by attribute-value pairs"

  $\rightarrow$ relational decision trees
- "Target function is *discrete* valued"

  $\rightarrow$ regression trees
- etc.

Top-Down Induction of Decision Trees

Main loop:

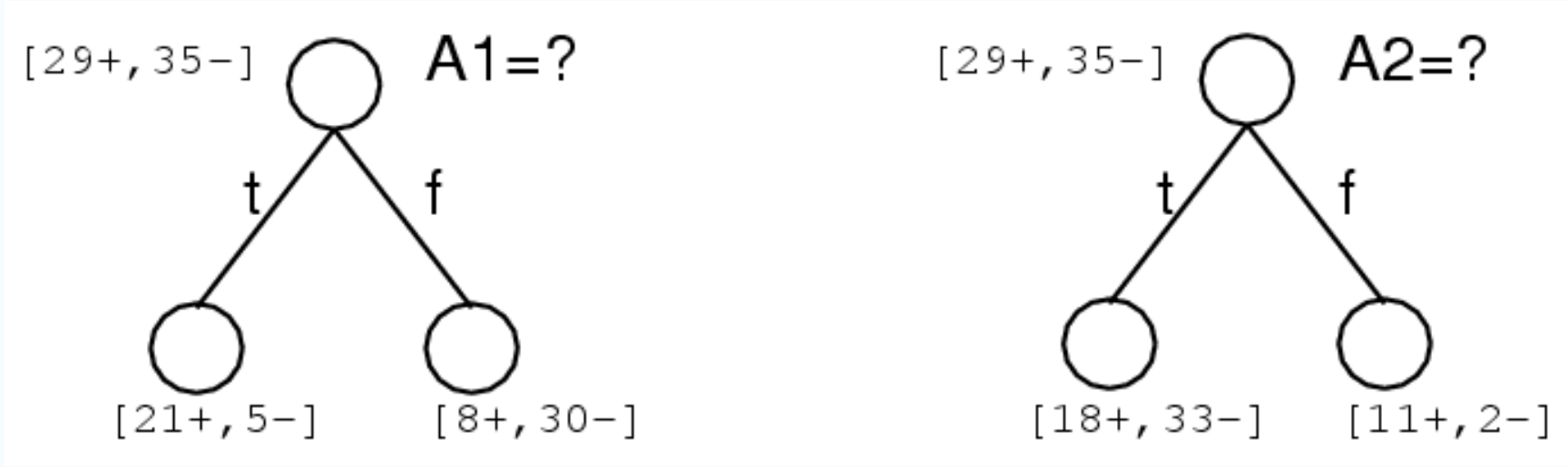
1. A $\leftarrow$ the "best" decision attribute for next node
2. Assign A as decision attribute for node
3. For each value of A, create new descendant of node
4. "Sort" training examples to leaf nodes
5. If training examples perfectly classified, then stop, else iterate over new leaf nodes and apply procedure recursively

Why Greedy Search?

- Early NP completeness results for decision tree construction
- However, there are (older and more recent) dynamic programming approaches to construct all decision under user-defined constraints (cf. constraint-based data mining)

Choosing Attributes: Entropy and Information Gain

Which Attribute is Best?



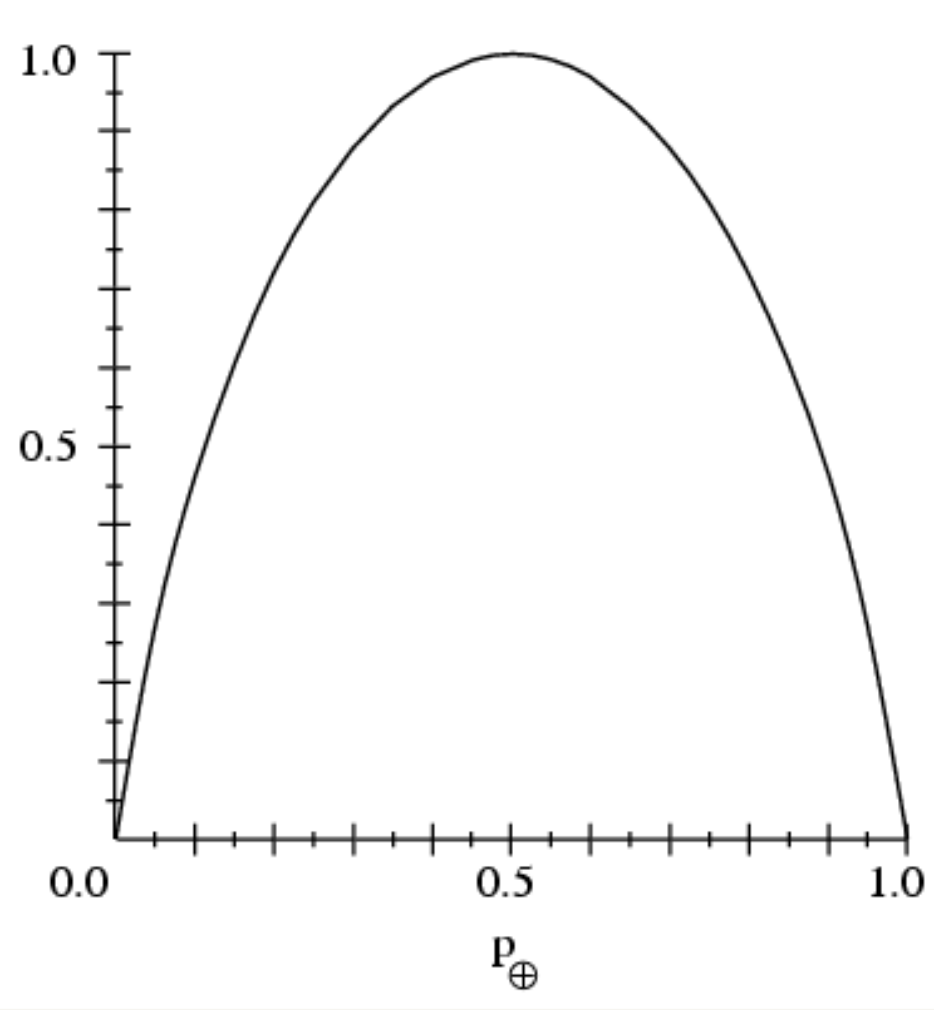
Evaluation of Splits by Information Gain

- Evaluation by so-called *information gain*
- Optimal length code of message of probability p: $-\log_2(p)$
- Expected number of bits needed to encode class (positive or negative) of random member of a set S:
  $\text{Entropy}(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-)$

Entropy

Entropy(S) =

- $p_+ \log_2(p_+)$
- $p_- \log_2(p_-)$



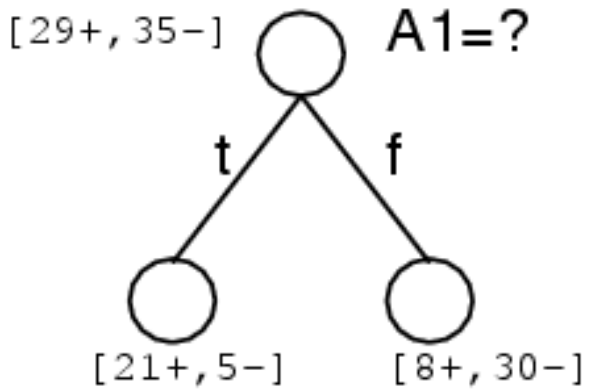
Evaluation of Splits by Information Gain

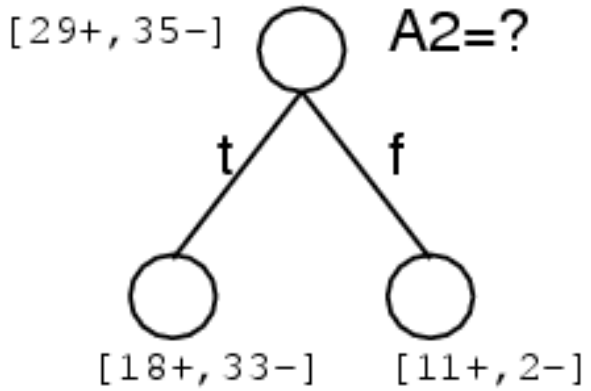
- Maximum of 1 for $p_+ = p_- = 0.5$
- Minimum of 0 for $p_+ = 1$, $p_- = 0$ or vice versa
- Expected reduction in entropy by splitting up S into S_t and S_f according to literal L
- Gain(S, L) =
  Entropy(S) –
  Entropy(S_t)|S_t| / |S| –
  Entropy(S_f)|S_f| / |S|

Evaluation of Splits by Information Gain

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

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





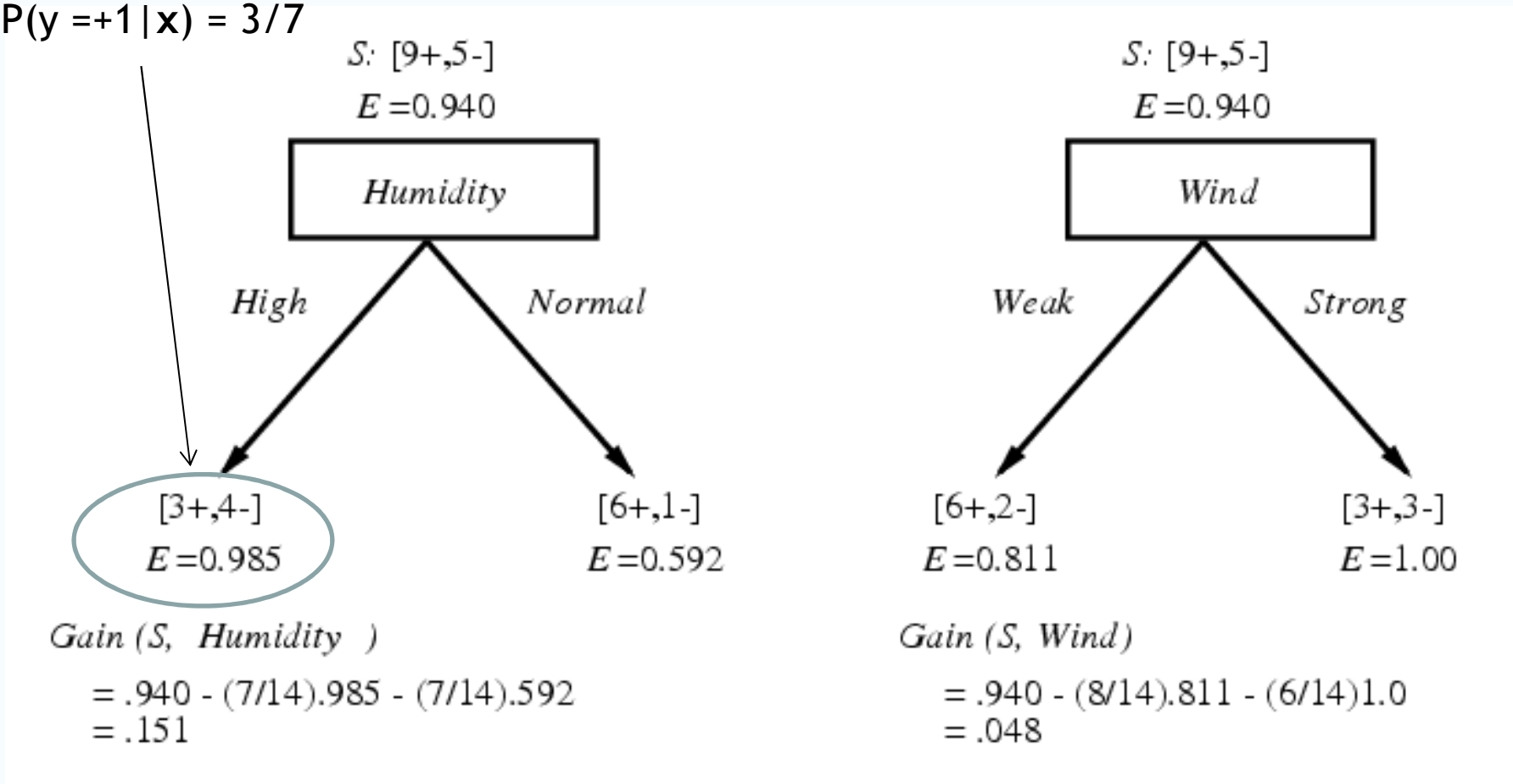
Example Dataset

Day	Outlook	Temp.	Hum.	Wind	PlayT.
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

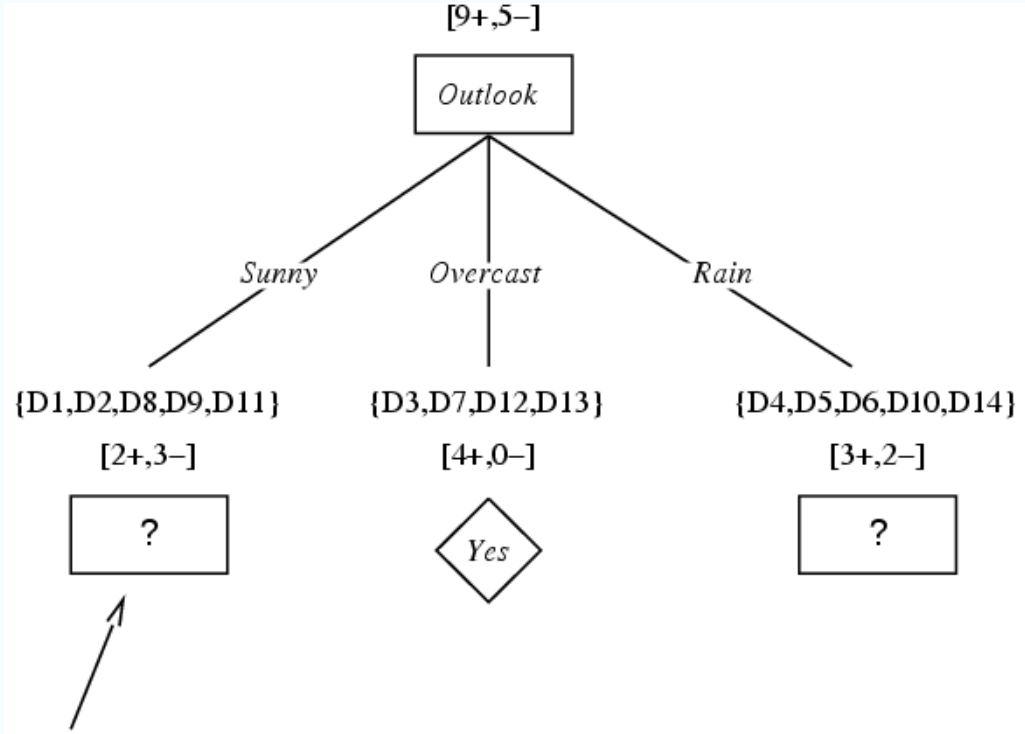
Example Calculation

Class probability estimates (!):

P(y =+1|**x**) = 3/7



Evaluating the Next Attribute



Which attribute should be tested here?

S_{sunny} = {D1,D2,D8,D9,D11}

Gain (S_{sunny}, *Humidity*) = .970 − (3/5) 0.0 − (2/5) 0.0 = .970

Gain (S_{sunny}, *Temperature*) = .970 − (2/5) 0.0 − (2/5) 1.0 − (1/5) 0.0

Gain (S_{sunny}, *Wind*) = .970 − (2/5) 1.0 − (3/5) .918 = .019