



Big Data Analytics

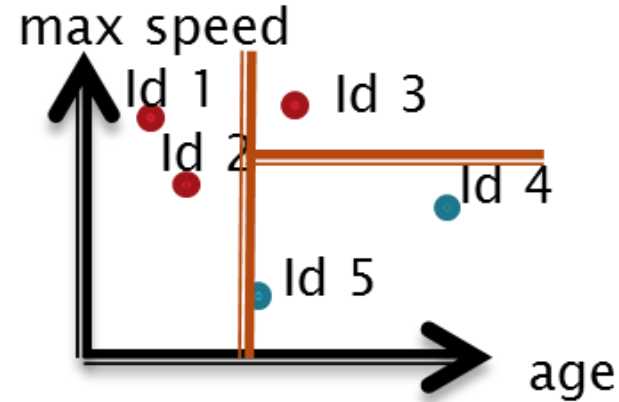
– Week 3: Decision Trees –

Knowledge Discovery and Data Mining
Prof. Dr. Emmanuel Müller

Chapter 5: Classification

Decision Tree Classifiers

- basic notions
- split strategies
- overfitting
- pruning of decision trees



Decision Tree Classifiers

Approximating discrete-valued target function

Learned function is represented as a tree:

- A flow-chart-like tree structure
- Internal node denotes a test on an attribute
- Branch represents an outcome of the test
- Leaf nodes represent class labels or class distribution

Learned tree can be transformed into IF-THEN rules

IF car_type = truck THEN risk = low

IF car_type ≠ truck AND age > 60 THEN risk = low

IF car_type ≠ truck AND age ≤ 60 THEN risk = high

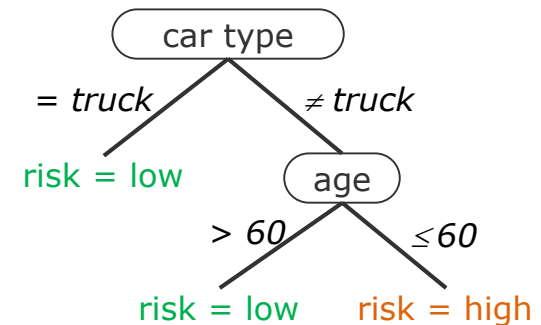
Advantages:

- Decision trees represent explicit knowledge
- Decision trees are intuitive to most users

training data

ID	age	car type	risk
1	23	family	high
2	17	sportive	high
3	43	sportive	high
4	68	family	low
5	32	truck	low

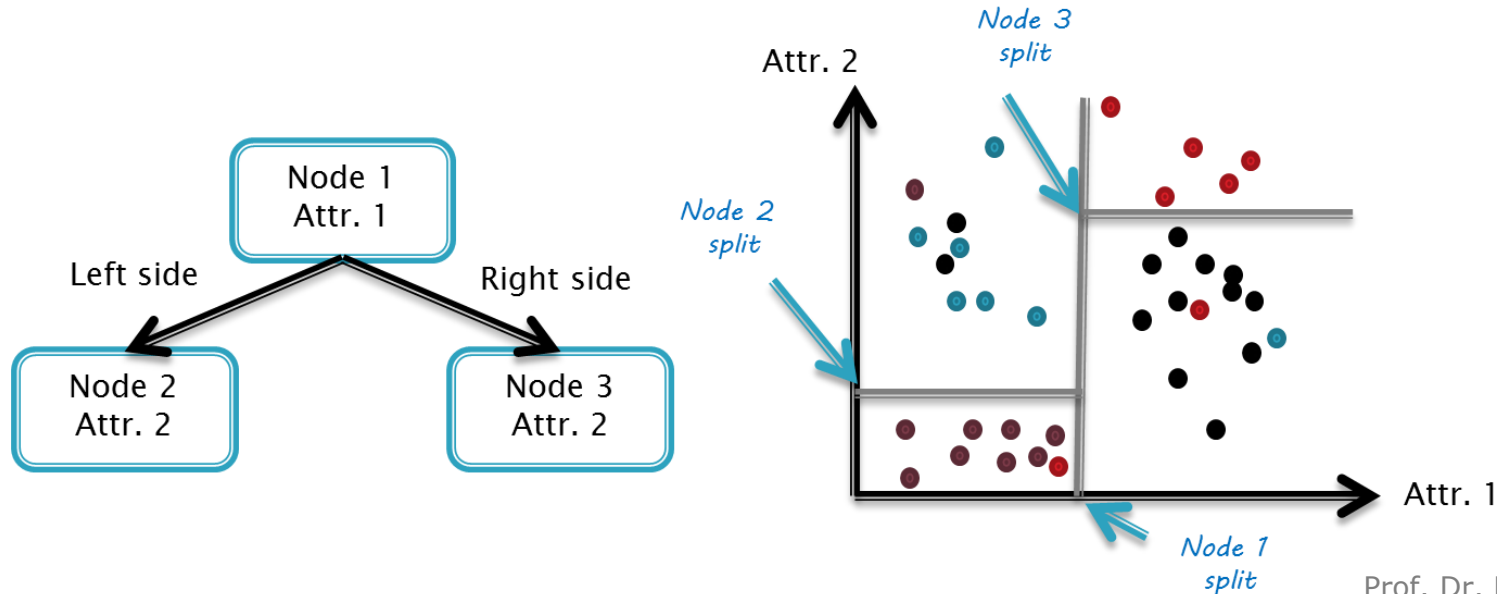
learned decision tree



Decision Tree Classifier: Splits

Each tree node defines an axis-parallel (d-1)-dimensional hyper plane, that splits the domain space

Goal: Find such splits which lead to as homogenous groups as possible



Decision Tree Classifiers: Basics

Decision tree generation (training phase) consists of two phases:

- Tree construction
 - At start, all the training examples are at the root
 - Partition examples recursively based on selected attributes
- Tree pruning
 - Identify and remove branches that reflect noise or outliers

Use of decision tree: classifying an unknown sample

- Traverse the tree and test the attribute values of the sample against the decision tree (simply follow path from the root to one of the leaf nodes)
- Assign the class label of the respective leaf to the query object

Algorithm for Decision Tree Construction

Basic algorithm (a greedy algorithm)

- Tree is created in a top-down recursive divide-and-conquer manner
- Attributes may be categorical or continuous-valued
- At start, all the training examples are assigned to the root node
- Recursively partition the examples at each node and push them down to the new nodes
 - Select test attributes and determine split points or split sets for the respective values on the basis of a heuristic or statistical measure (split strategy, e.g., information gain)

Conditions for stopping partitioning

- All samples for a given node belong to the same class
- There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
- There are no samples left

Algorithm for Decision Tree Construction

Most algorithms are versions of this basic algorithm (greedy, top-down) ID3[Q86] or C4.5[Q96]

ID3(*Examples*, *TargetAttr*, *Attributes*) //specialized to learn boolean-valued functions

Create a *Root* node for the tree;

If all *Examples* are positive, return *Root* with label = + ;

If all *Examples* are negative, return *Root* with label = - ;

If *Attributes*= $\emptyset$, return *Root* with label = most common value of *TargetAttr* in *Examples*;

Else

A=the 'best' decision attribute for next node

 how to determine the 'best' attribute ?

Assign *A* as decision attribute for *Root*

For each possible value v_i of *A*:

 how to split the possible values ?

Generate branch corresponding to test $A = v_i$;

$Examples_{v_i}$ = examples that have value v_i for *A*;

If $Examples_{v_i} = \emptyset$, add leaf node with label = most common value of *TargetAttr* in *Examples*;

Else add subtree ID3($Examples_{v_i}$, *TargetAttr*, $Attributes \setminus \{A\}$);

[Q86] J.R. Quinlan. *Induction of decision trees*. Machine Learnin, 1(1), pages 81-106, 1986.

[Q96] J. R. Quinlan. *Bagging, boosting, and C4.5*.

In Proc. 13th Natl. Conf. on Artificial Intelligence (AAAI'96), pages 725-730, 1996.

Example: Decision for „playing_tennis“

Query: How about playing tennis today?

Training data:

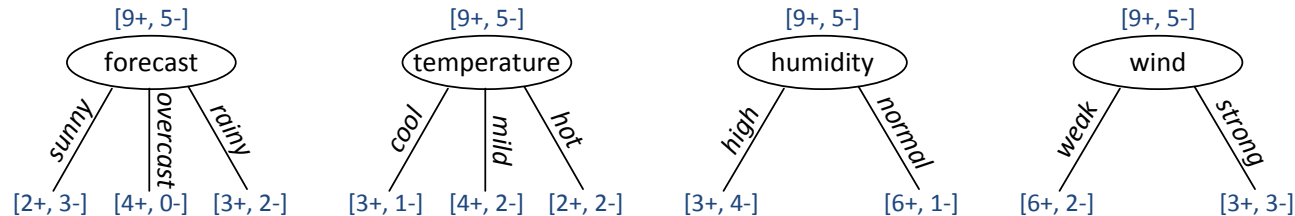
day	forecast	temperature	humidity	wind	tennis decision
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rainy	mild	high	strong	no

Build decision tree ...

Split Strategies: Quality of Splits

Given

- a set T of training objects
- a (disjoint, complete) partitioning $T_1, T_2, \dots, T_m$ of T
- the relative frequencies p_i of class c_i in T and in the partition $T_1, \dots, T_m$



Searched

- a measure for the heterogeneity of a set S of training objects with respect to the class membership
- a split of T into partitions $T_1, T_2, \dots, T_m$ such that the heterogeneity is minimized

Proposals: Information gain, Gini index, Misclassification error

Attribute Selection Measures: Information Gain

used in ID3 / C4.5

Entropy

- minimum number of bits to encode a message that contains the class label of a random training object
- the entropy of a set T of training objects is defined as follows:

$$\text{entropy}(T) = - \sum_{i=1}^k p_i \cdot \log_2 p_i$$

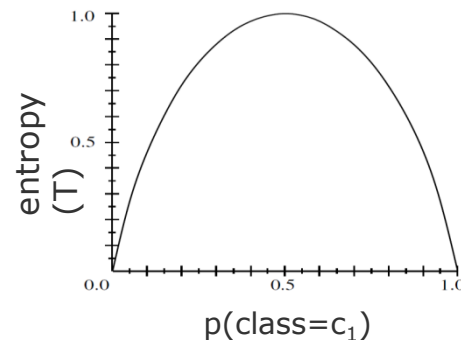
for k classes c_i with frequencies p_i

- $\text{entropy}(T) = 0$ if $p_i = 1$ for any class c_i
- $\text{entropy}(T) = 1$ if there are $k = 2$ classes with $p_i = 1/2$ for each i

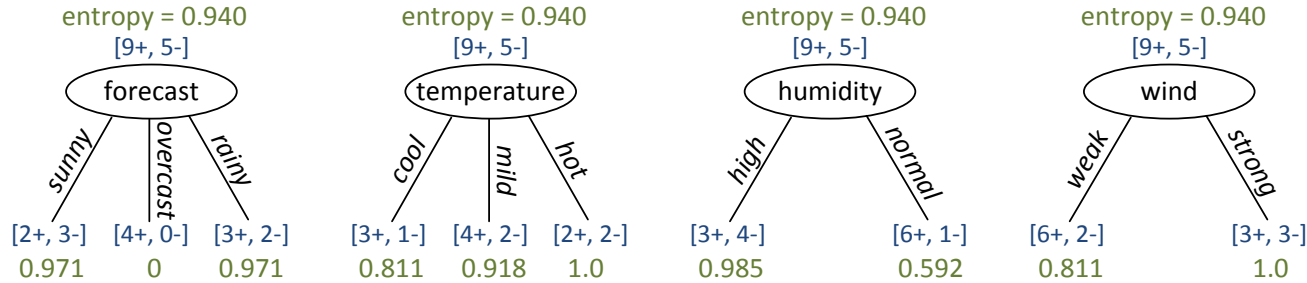
Let A be the attribute that induced the partitioning $T_1, T_2, \dots, T_m$ of T . The information gain of attribute A w.r.t. T is defined as follows:

$$\text{information gain}(T, A) = \text{entropy}(T) - \sum_{i=1}^m \frac{|T_i|}{|T|} \cdot \text{entropy}(T_i)$$

for two
classes:



Attribute Selection: (Information Gain)



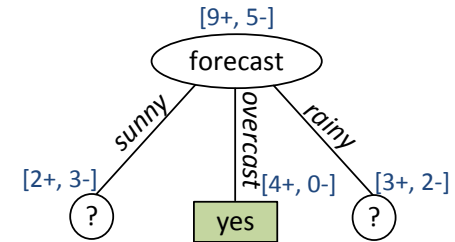
$$\text{information gain}(T, \text{forecast}) = 0.94 - \frac{5}{14} \cdot 0.971 - \frac{4}{14} \cdot 0 - \frac{5}{14} \cdot 0.971 = 0.246$$

$$\text{information gain}(T, \text{temperature}) = 0.94 - \frac{4}{14} \cdot 0.811 - \frac{6}{14} \cdot 0.918 - \frac{4}{14} \cdot 1 = 0.029$$

$$\text{information gain}(T, \text{humidity}) = 0.94 - \frac{7}{14} \cdot 0.985 - \frac{7}{14} \cdot 0.592 = 0.151$$

$$\text{information gain}(T, \text{wind}) = 0.94 - \frac{8}{14} \cdot 0.811 - \frac{6}{14} \cdot 1.0 = 0.048$$

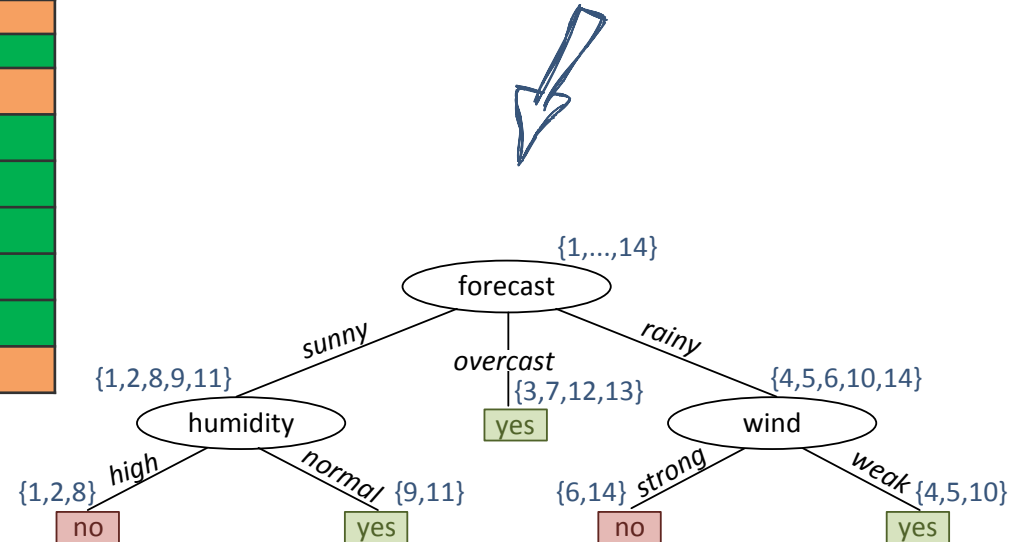
Result: 'forecast' yields the highest information gain



Example: Decision for „playing_tennis“

day	forecast	temperature	humidity	wind	decision
1	sunny	hot	high	weak	no
2	sunny	hot	high	strong	no
3	overcast	hot	high	weak	yes
4	rainy	mild	high	weak	yes
5	rainy	cool	normal	weak	yes
6	rainy	cool	normal	strong	no
7	overcast	cool	normal	strong	yes
8	sunny	mild	high	weak	no
9	sunny	cool	normal	weak	yes
10	rainy	mild	normal	weak	yes
11	sunny	mild	normal	strong	yes
12	overcast	mild	high	strong	yes
13	overcast	hot	normal	weak	yes
14	rainy	mild	high	strong	no

final decision tree



Attribute Selection Measures: Gini Index

Used in IBM's Intelligent Miner

The *Gini index* for a set T of training objects is defined as follows

$$gini(T) = 1 - \sum_{j=1}^k p_j^2 \quad \text{for } k \text{ classes } c_j \text{ with frequencies } p_j$$

- small value of Gini index $\Leftrightarrow$ low heterogeneity
- large value of Gini index $\Leftrightarrow$ high heterogeneity

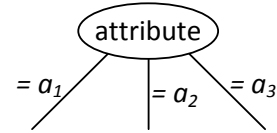
Let A be the attribute that induced the partitioning $T_1, T_2, \dots, T_m$ of T . The *Gini index* of attribute A w.r.t. T is defined as follows:

$$gini_A(T) = \sum_{i=1}^m \frac{|T_i|}{|T|} \cdot gini(T_i)$$

Split Strategies: Types of Splits

Categorical attributes

- split criteria based on equality „ $attribute = a$ “
- based on subset relationships „ $attribute \in set$ “
→ many possible choices (subsets)
 - Choose the best split according to, e.g., gini index



Numerical attributes

- split criteria of the form „ $attribute < a$ “
→ many possible choices for the split point
 - One approach: order test samples w.r.t. their attribute value; consider every mean value between two adjacent samples as possible split point; choose best one according to, e.g., gini index
- partition the attribute value into a discrete set of intervals (Binning)

