CHAIR FOR BIOINFORMATICS
AND INFORMATION MINING

Inference of Decision Graphs using the Minimum Message Length Principle

Tobias Witt

Chair for Bioinformatics and Information Mining
Department of Computer and Information Science
University of Konstanz, Germany
tobias.witt@uni-konstanz.de

Universität Konstanz

Agenda

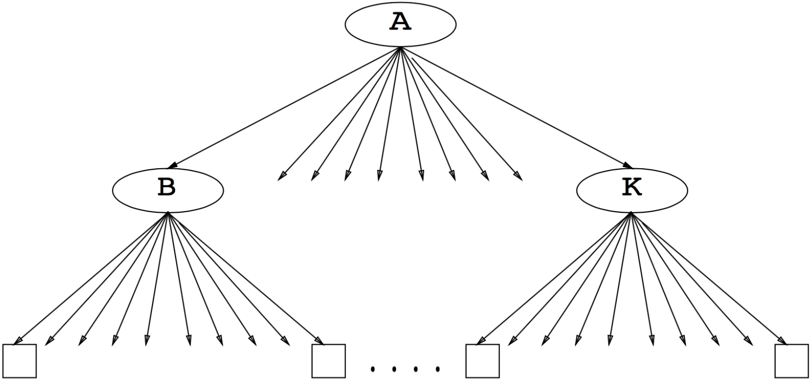
1. Problems of Decision Trees
2. Decision Graphs
3. Minimum Message Length Principle for Decision Graphs
4. Accuracy Tests

Decision Tree

- A Decision Tree represents a *set of decision rules*.
- Each path from the root to a leaf constitutes one rule.
- The tree can depict arbitrary concepts (disjunction of conjunctions).
- The tree representation might be problematic for two reasons:
  1. Quick fragmentation
  2. Duplication of subtrees

Fragmentation Problem

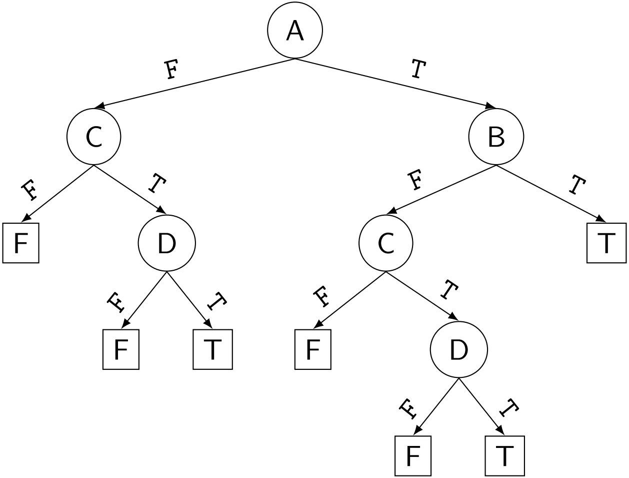
500 training items



100 leaves with an average of 5 training items

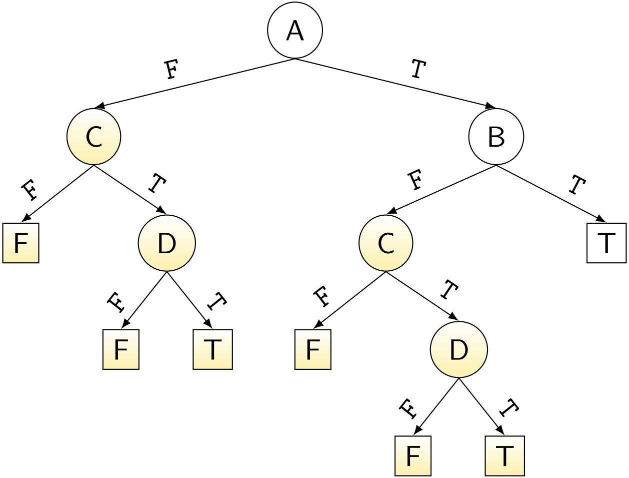
Replication Problem

Decision Tree representation of $(A \wedge B) \vee (C \wedge D)$



Replication Problem

Decision Tree representation of $(A \wedge B) \vee (C \wedge D)$



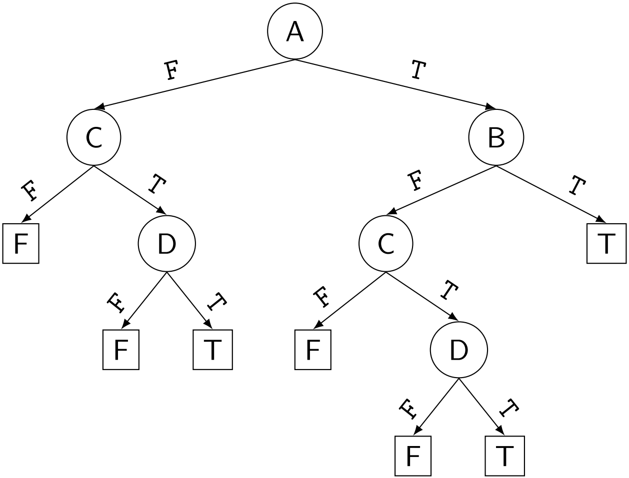
Decision Graph

- Potentially large decision trees with few observations in single nodes.
- Consequences:
  1. More data needed to learn the underlying concept.
  2. Decision rules are harder to interpret.
- Idea: Instead of a tree, use a *directed acyclic graph* to represent decision rules $\Rightarrow$ **Decision Graph**[1]
- Representation allows for *Joins* (two nodes can have a common child).
- Permits a more efficient representation of disjunctive rules.

[1] Decision Graphs were introduced by Oliver (1993).

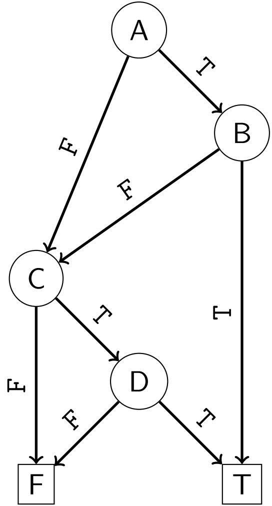
Decision Graph

Decision Tree representation of $(A \wedge B) \vee (C \wedge D)$



Decision Graph

Decision Graph representation of $(A \wedge B) \vee (C \wedge D)$



Decision Graph

Grow procedure: Starting with a single leaf node, the follwing steps are repeated until no further improvements can be achieved.

1. For each leaf determine the attribute which it should be split on and perform a tentative split.
2. For each combination of leaves perform a tentative join.
3. Choose the "best" alteration (from Step 1 or Step 2) and perform it on the graph. Do nothing if no alteration improves the graph.

⇒ Apply **Minimum Message Length Principle** (see Wallace, 2005) to heuristically search for the best Decision Graph.

Minimum Message Length

- Communication problem: Person 1 wants to transmit the missing class column to Person 2 *using as few bits as possible*.

Person 1

	Outlook	Temp.	Windy	PlayTennis
1	sunny	85	false	yes
2	sunny	80	true	no
3	overcast	83	false	yes
4	rain	70	false	yes
5	rain	68	false	yes
6	rain	65	true	no

Person 2

	Outlook	Temp.	Windy	PlayTennis
1	sunny	85	false	
2	sunny	80	true	
3	overcast	83	false	
4	rain	70	false	
5	rain	68	false	
6	rain	65	true	

- If the class depends on the attributes, Person 1 might save bits by transmiting a model which Person 2 can use to deduce the class column from the attributes.

Minimum Message Length

- **MML Principle:** Given a set of data, the *best* model is the one which requires the fewest possible bits to describe the data, i.e., the model allowing for the shortest message.
- The message consists of
  (1) a description of the theory/hypothesis $h \in H$, and
  (2) the data D explained with the help of h.
- *Classical Information Theory*: Using an optimal code, the length of the full message (h, D) is

$$\begin{aligned} \mathrm{length}(h, D) &= -\log_2(P(h, D)) \\ &= -\log_2\left(P(h)P(D|h)\right) \end{aligned}$$

Minimum Message Length

- Minimizing $-\log_2(P(h)P(D|h))$ is equal to maximizing $P(h|D)$

$$\arg\max_{h \in H} P(h|D) = \arg\max_{h \in H} \frac{P(h)P(D|h)}{P(D)} = \arg\max_{h \in H} P(h)P(D|h)$$

$$= \arg\min_{h \in H} -\log_2(P(h)P(D|h))$$

- Tradeoff between model complexity and goodness of fit

$$\begin{aligned} \text{length}(h, D) &= -\log_2(P(h)P(D|h)) \\ &= -\log_2(P(h)) - \log_2(P(D|h)) \\ &= \text{length}(h) + \text{length}(D|h) \end{aligned}$$

MML Principle for Decision Trees

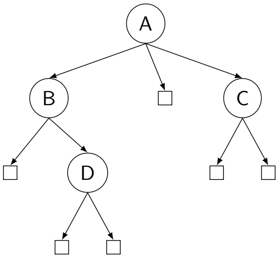
- Wallace and Patrick (1993) propose a coding scheme to use the MML Principle for Decision Trees.[2]
- To transmit data using a decision tree, the message must contain
  1. the **structure** of the tree,
  2. the class **prediction** and the **exceptions** for each leaf node.

[2] Their work is heavily based on Quinlan and Rivest (1989).

MML Principle for Decision Trees

Structure Message

- Remember: The receiver knows the attributes.



`1 A 1 B 0 1 D 0 0 0 1 C 3.5 0 0`

- With optimal code: $\text{length}(s) = -\log(P(s))$ for structure $s \in S$
- Wallace and Patrick (1993) propose procedure to calculate $P(s)$.

MML Principle for Decision Trees

Category message

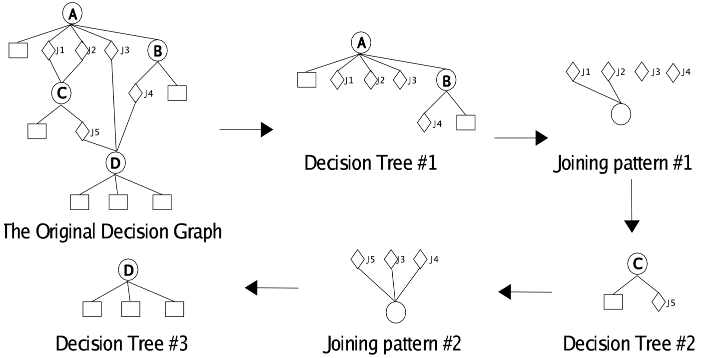
- Transmit the distribution of class values within each leaf node.
- Example: `[1 2 2 1 2 0 2 0 2 2]`, $p_0 = 0.2$, $p_1 = 0.2$, $p_2 = 0.6$
- Assume generalized symmetric Beta prior over unknown class probabilities p_m for M classes

$$f(p_1, \ldots, p_m) = \prod_m p_m^{(\alpha-1)}$$

- $\alpha = 1$: Uniform prior (all distributions equally likely).
- $\alpha < 1$: Greater weight is placed on pure distributions.
- Each instance is encoded with length $-\log_2(q)$, where q is the updated (expected) probability of the element's class after having observed the previous elements (**incremental code**).

MML Principle for Decision Graphs

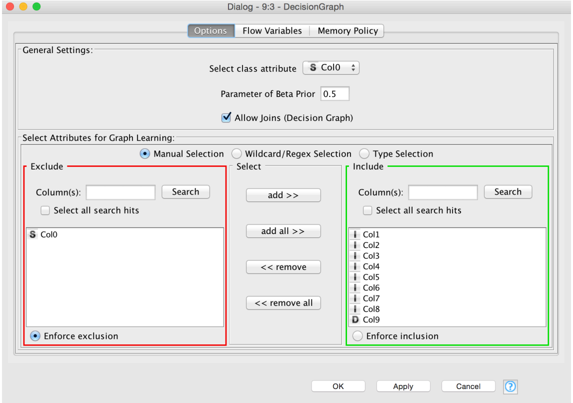
- Tan and Dowe (2003): Decompose Decision Graph into a sequence of decision trees to apply the scheme by Wallace and Patrick (1993)



- Complete Message = Structure Messages + Category Messages + Joining Pattern Messages

Own Implementation in KNIME

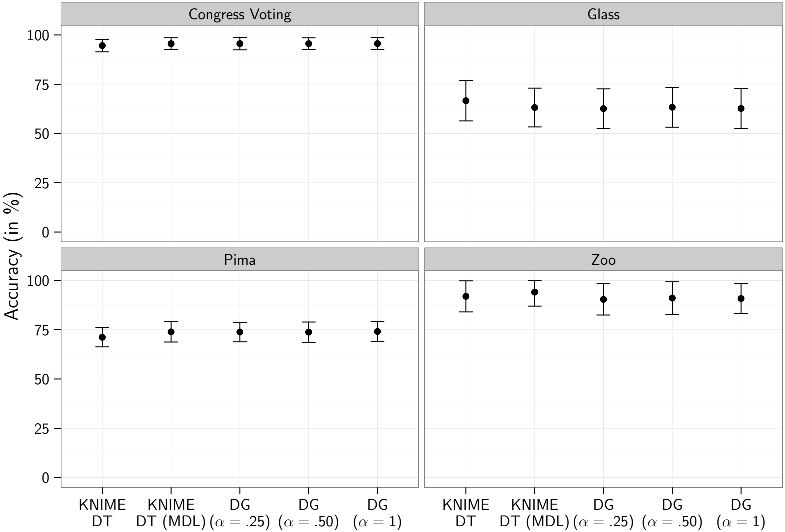
- Learner and predictor combined in one node.
- Two input ports for training and test data.
- Node handles categorical and continuous attributes.



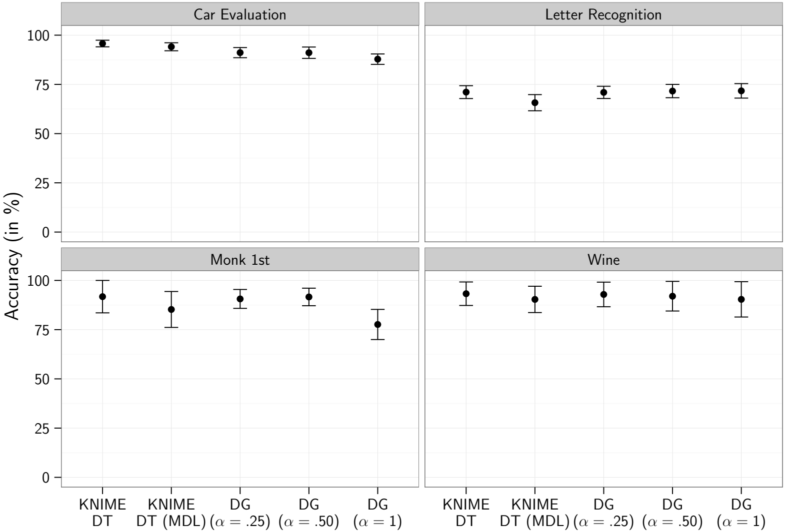
Accuracy Tests

- 10 fold cross-validation repeated 50 times.
- Data Sets from the UCI Machine Learning Repository:
  - Pima ($N = 768$)
  - Congress Voting ($N = 435$)
  - Zoo ($N = 101$)
  - Glass ($N = 214$)
  - Letter Recognition ($N = 20,000$)
  - Car ($N = 1728$)
  - Wine ($N = 178$)
  - Monk 1st ($N = 432$)

Accuracy Tests



Accuracy Tests



Accuracy Tests

- The predictive power of the DG-algorithm is similiar to the Decision Tree algorithms implemented in KNIME.
- Overall, the Decision Tree algorithm with reduced error pruning performs best.
- The join operator in the DG algorithm is used rarely.
- Joins are predominantly performed once no further split can reduce the message length.
- Plans for the paper:
  - More testing with different data sets.
  - Different encoding of the joining pattern message.

Literature

Oliver, Jonathan J. 1993. Decision Graphs - An Extension of Decision Trees. In *Proceedings of the Fourth International Workshop on Artificial Intelligence and Statistics*. pp. 343–350.

Quinlan, John Ross and Ronald L. Rivest. 1989. "Inferring Decision Trees Using the Minimum Description Length Principle." *Information and Computation* 80(1989):227–248.

Tan, Peter J. and David L. Dowe. 2003. "MML Inference of Decision Graphs with Multi-Way Joins and Dynamic Attributes." *Proceedings of the 16th Australian Joint Conference on Artificial Intelligence* pp. 269–281.

Wallace, C. S. 2005. *Statistical and Inductive Inference by Minimum Message Length*. Springer-Verlag New York.

Wallace, C. S. and J. D. Patrick. 1993. "Coding Decision Trees." *Machine Learning* 11(1):7–22.