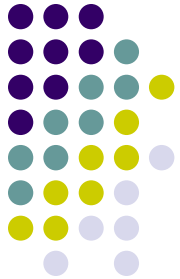
Deriving rules from data

Decision Trees

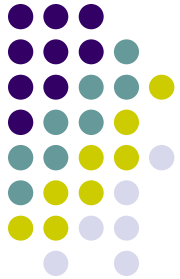
a.j.m.m (ton) weijters

Knowledge discovery and Data Mining



- **Knowledge discovery in databases (KDD)** is the process of identifying valid, novel, potentially useful, and ultimately understandable patterns or models in data. **Data mining(DM)** is a step in the knowledge discovery process consisting of particular data mining algorithms that, under some acceptable computational efficiency limitations, find patterns or models in data.

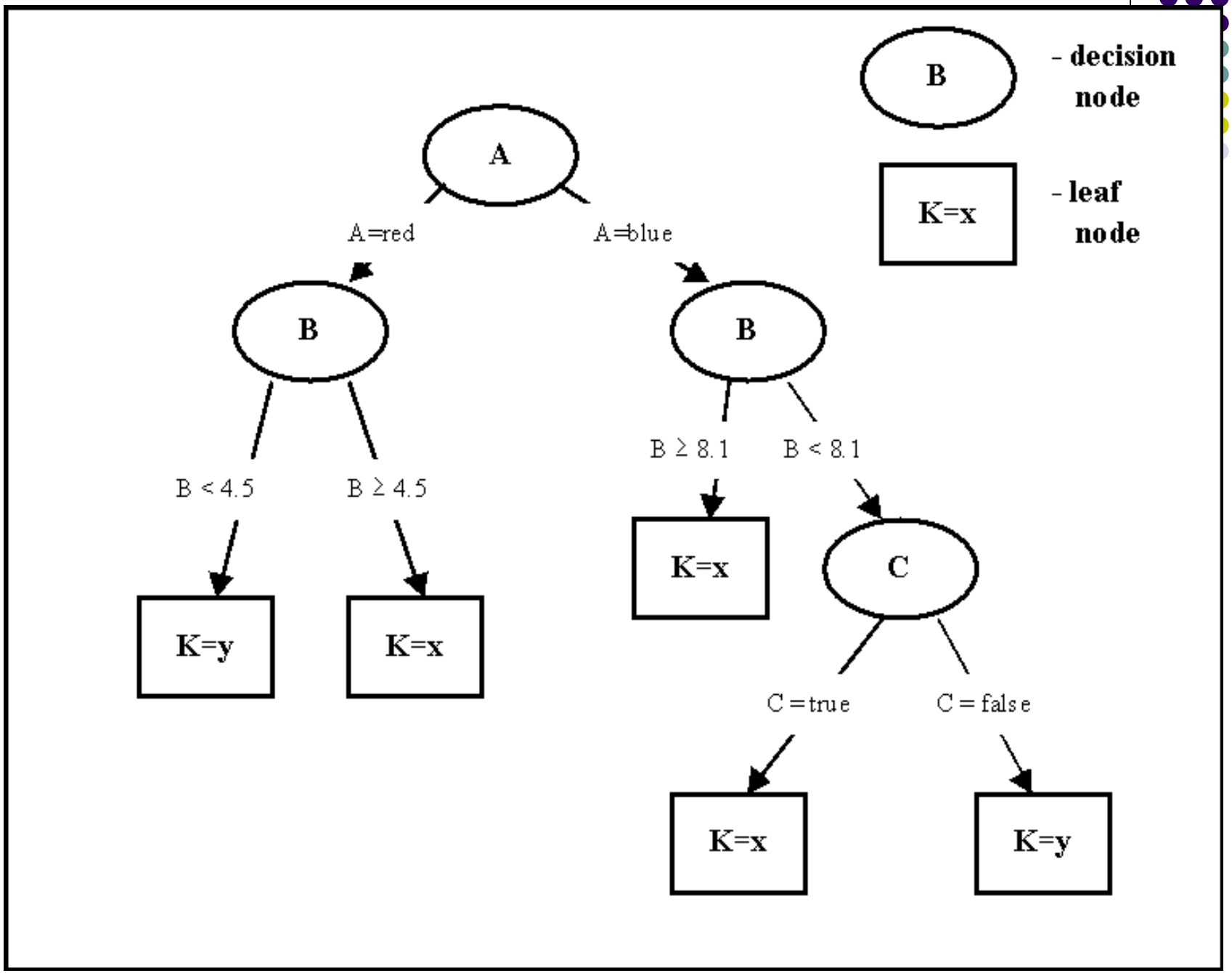
Decision Trees



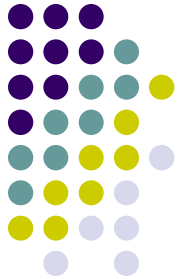
Decision trees are powerful and popular tools for classification and prediction. The attractiveness of decision trees is due to the fact that, in contrast to other data mining (ML) techniques, decision trees represent *rules*. Rules can readily be expressed so that humans can understand them or even directly used in a database access language like SQL so that records falling into a particular category may be retrieved

What is a decision tree ?

- *Decision tree* is a classifier in the form of a tree structure (see Figure 1), where each node is either:
- a *leaf node* - indicates the value of the target attribute (class) of examples, or
- a *decision node* - specifies some test to be carried out on a single attribute-value, with one branch and sub-tree for each possible outcome of the test.
- A decision tree can be used to classify an example by starting at the root of the tree and moving through it until a leaf node, which provides the classification of the instance.

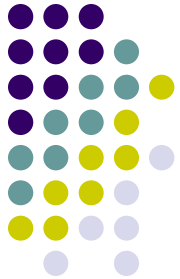

B
- decision node
K=x
- leaf node
A
A=red
A=blue
B
B
B < 4.5
B ≥ 4.5
B ≥ 8.1
B < 8.1
K=y
K=x
K=x
C
C = true
C = false
K=x
K=y

Requirements



- *Attribute-value description: object or case must be expressible in terms of a fixed collection of properties or attributes.*
- *Predefined classes (target attribute values): The categories to which examples are to be assigned must have been established beforehand (supervised data).*
- *Discrete classes: A case does or does not belong to a particular class, and there must be more cases than classes.*
- *Sufficient data: Usually hundreds or even thousands of training cases.*

Constructing decision trees



- searches through the attributes of the training instances and extracts the attribute that best separates the given examples.
- If the attribute perfectly classifies the training sets then stop; otherwise recursively operates on the next "best" attribute.
- The algorithm picks the best attribute and never looks back to reconsider earlier choices.

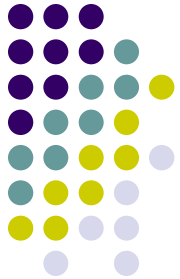
Illustration *(10 learning examples):*

	Hair	Length	Weight	Suntan cream	Burned
1	blond	medium	light	yes	no
2	blond	medium	light	no	yes
3	red	long	light	yes	yes
4	brown	medium	heavy	yes	no
5	blond	long	medium	yes	no
6	brown	long	light	no	no
7	red	small	heavy	no	yes
8	brown	long	light	yes	no
9	blond	medium	heavy	no	yes
10	brown	small	heavy	no	no

New (test) examples:

1	red	medium	light	yes	yes
2	blond	medium	medium	no	yes
3	brown	small	light	yes	no

Search for the best split



- Entropy and Information Gain
- Chi Square test

Entropy (E):

If S is a collection of examples and the target attribute can take c different values, then the entropy of a set S relative to this c-wise classification is defined as

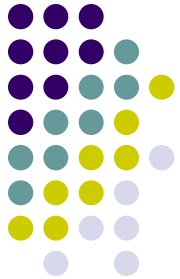
$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

with p_i is the proportion of S belonging to class i.

```
---------- 0 log 0 + 1 log 1 = 0
++--------- 2/10 log 2/10 + 8/10 log 8/10 =
                -0.464 + -0.258 = 0.722
++++------ 5/10 log 5/10 + 5/10 log 5/10 = 1
+++++++++- 0.722
++++++++++ 0
```

++++---XXXX

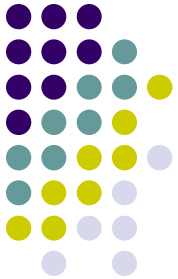
Information Gain (IG):



The information gain of an attribute A, relative to a collection of examples S (notation (Gain(S,A)) is defined as

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

Example



+ + + + - - - - - -

***E(D)* = - (6/10 ^{2}log 6/10 + 4/10 ^{2}log 4/10) = 0.442 + 0.529 = 0.97**

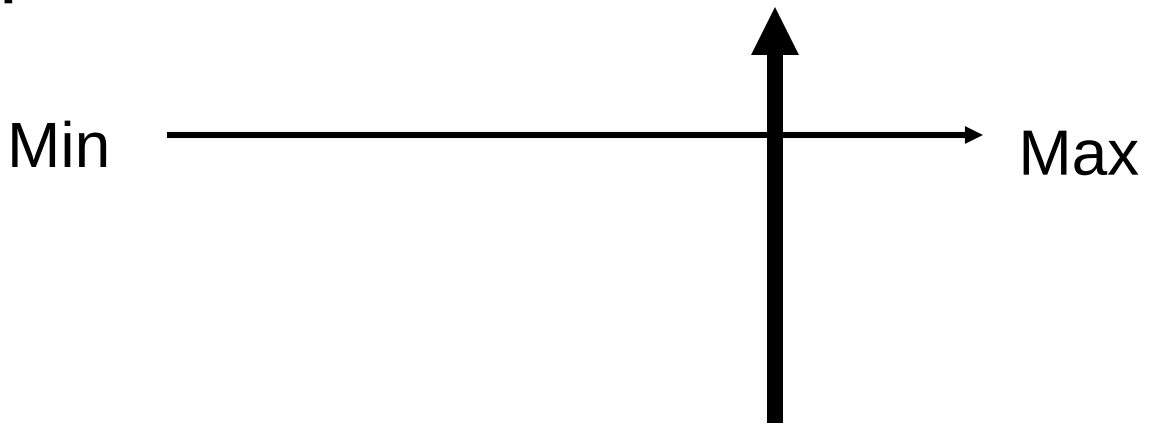
- Hair: 0.57 (+ +), (+ + - -), (- - - -)
- Length: 0.20 (+ -), (+ + - -), (+ - - -)
- Weight: 0.07 (+ + - - -), (-), (+ + - -)
- Suntan cream: 0.18 (+ + + - -), (+ - - - -)

- 0.97 – 2/10*0 + 4/10*1 + 4/10*0 = 0.57

Decision trees and continues values

- look for a binary split with max IG

- example: - - + + - - + + + + + - - - - - - - - -



Chi Square homogenous test

		smoker	**no smoker**	Total
man	Observed	46	154	200
	Expect	53.33	146.67	
woman	Observed	34	66	100
	Expect	26.67	73.33	
	Total	80	220	300
Overall %		0.27	0.73	

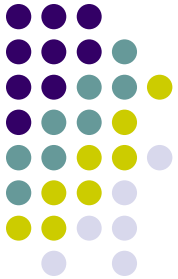
Alpha=0.05, Degrees of Freedom 2 (rows)-1 x 2 (classes)-1=1
Chi-Square(0.05, 1)=3.84

		smoker	no smoker	Total
man	Observed	46	154	200
	Expect	53.33	146.67	
woman	Observed	34	66	100
	Expect	26.67	73.33	
	Total	80	220	300
Overall %		0.27	0.73	

ChiSq = $(46-53.3)^2/53.3 + (154-146.7)^2/146.7 + (34-26.7)^2/26.7 + (66-73.3)^2/73.3 = 4.125 > 3.84$

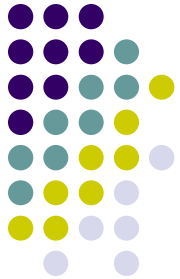
Conclusion: F M is 0.05 significant information for smoking behaviour!

From decision tree to rules



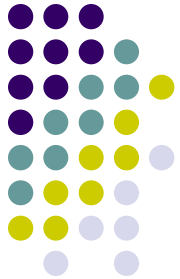
- From decision tree to IF-THEN-rules:
  - IF hair=red THEN burning=yes
  - IF hair=brown THEN burning=no
  - IF hair=blond AND suntan_cream=yes THEN burning=no
  - IF hair=blond AND suntan_cream=no THEN burning=yes
- Pruning

The strengths of decision trees



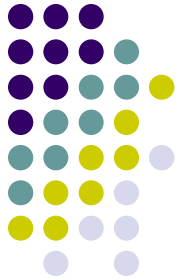
- Decision trees are able to generate understandable rules.
- Decision trees perform classification without requiring much computation.
- Decision trees are able to handle both continuous and categorical variables.
- Decision trees can handle non-linear separations'.
- Decision trees provide a clear indication of which fields are most important for prediction or classification.

The weaknesses of decision trees



- Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
- Decision trees are prone to errors in classification problems with many classes and relatively small number of training examples.
- Most decision-tree algorithms only examine a single field at a time. Problems with informative combinations of features.

AlphaMiner



- Public domain (Weka)
- Heuristics
  - Start with a complex tree (min size of a leaf node)
  - Pruning during rule building (confidence threshold or n-fold CV)

Conclusions

- Recursive tree construction is a robust method (a good first start!)
- Can handle non linear relations
- It is clear what you are doing and it gives you insight in the data
- Clear rules with a good indication of the reliability of the rules
- AlphaMiner is a useful tree construction tool