Submission until: **03.05.** (late submissions will get a deduction)
Discussion on: 05.05.
Submission as upload to your groups stud.IP folder as *groupNumber_sheet2.zip*

Assignment 1 *(Decision Trees (2p))*

Build/draw the decision trees for the following boolean functions:

(a) $A \oplus \neg B$ ($\oplus$ = xor)

(b) $(A \wedge B) \vee (\neg B \wedge C)$

(c) $(A \rightarrow B) \vee (A \rightarrow \neg C)$

(d) $(A \wedge B) \vee (\neg A \wedge C) \vee D$

Assignment 2 *(Entropy and information gain - 8p)*

Table 1: Attributes and their possible values

genre	main-character	has_ninjas
action,romance,comedy	male,female	true,false

Table 2: training examples

Nr.	genre	main-character	has_ninjas	watch
1	action	male	true	no
2	romance	male	true	yes
3	action	female	true	yes
4	comedy	female	true	yes
5	romance	female	false	no

(a) (*4*p) Consider the five training examples from Table 2. Build the root node of a decision tree from these training examples.
To do this, you calculate the information gain on all three distinct attributes (genre, main-character, has_ninjas) to decide which one would be the best choice for the root node (the one with the largest gain).
The information gain is given as

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

The Entropy is given as

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

S_v is the subset of S for which attribute A has value v.
Example for attribute main-character:
$S_m \leftarrow [1+, 1-]$, $|S_m| = 2$
$S_f \leftarrow [2+, 1-]$, $|S_f| = 3$

Provide all detailed calculations and the result.

(b) (*2*p) Perform the same calculation as in a) but use the gain ratio instead of the information gain. Does the result for the root node change?

$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInformation(S, A)}$, with

$SplitInformation(S, A) = -\sum_{v \in Values(A)} \frac{|S_v|}{|S|} log_2 \frac{|S_v|}{|S|}$

(c) (*2*p) Let's assume the root node is a node which checks the value of the attribute *has_ninjas*. Calculate the next level of the decision tree using the information gain.

Programming Exercises

For the following tasks you can use either Matlab or Python. Only use builtin functions where they are explicitly permitted. Basic functions for file handling, array creation and manipulation as well as plotting are of course excluded from this regulations. For Python users this covers the use of the following modules:

1. *scipy.io* for handling .mat files
2. *numpy* for array creation/manipulation
3. *matplotlib.pyplot* for plotting

One last advice: Do NOT copy code from external sources and submit it as your own. If a group should happen to submit such code all group members will receive a serious deduction of points.

Assignment 3 *(Decision Trees (5p))*

Use builtin functions to solve this task. For Matlab have a look at the *classregtree* function. Python users should make use of the *DecisionTreeClassifier* class from the *scikit-learn* module, as well as *pydot* for plotting.

(a) (*2*p) Calculate a decision tree on the Iris data set (iris.mat).

(b) (*1*p) Perform a 3-fold cross-validation on the data set.

(c) (*2*p) Calculate the errors for the test classification. Display the best and worst decision tree.

Assignment 4 *(Z-test (5p))*

Given the data in *zPoints.mat*, implement a function that performs the Rosner test.

(a) (*3*p) using the mean.

(b) (*1*p) using the median.

Use 3.0 as value for the threshold. Plot the data points and highlight those that would be/are removed (*1*p).