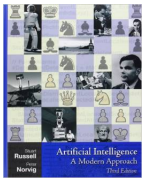
# TDT4171 Artificial Intelligence Methods

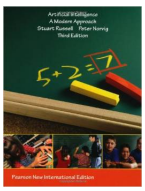
Exercise 4

March 3, 2016

- **Delivery deadline: April 21, 2016** by 8:00 a.m.
- Required reading for this assignment: Chapter 18.3.
- Deliver your solution on *It's Learning*.
- Students may work in groups of two.
- Please upload your report as a single **PDF file**, and pack everything else into an archive (zip, rar, gz, ...). Please **do not** put the pdf into the archive, but upload two separate files: pdf + archive. If you use Word, you can convert to pdf on the IDI terminal servers. If you want to use plain ASCII, then you should also convert that to PDF (e.g., by typing in Word and following the procedure just mentioned).
- This homework counts up to 3% of the final grade.
- The homework is graded on a pass/fail basis. A pass grade will only be given when a decent attempt has been made to solve each question in the exercise.
- Cribbing from other students (*"koking"*) is not accepted, and if detected will lead to the assignment being failed.
- The textbook, *Artificial Intelligence: A Modern Approach, 3rd edition*, exists in two versions:




"Blue version"



"Green version"

References to chapters, figures and pages in the textbook are given as 1.2.3 / 2.3.4, for the blue and the green version respectively, whenever they differ.

## Introduction

In this exercise you will implement a decision tree learning algorithm. Please make sure to read the exercise completely before starting the implementation.

You will implement the decision tree learning algorithm (pseudo-code in Figure 18.5 in the textbook, page 702 / 713). You can do this using the programming language you want, but make sure to deliver your source file, a print-out of the results, and appropriate discussion. Your source-files should be ready to compile; any non-standard dependencies must be specified and supplied. **Note also that just submitting your source code is not a sufficient answer - make sure to answer specific questions in the text!**

## Tasks

You should implement two different versions of IMPORTANCE function (used by pseudo-code in Figure 18.5):

1. Allocating a random number as importance to each attribute.
2. Define Importance as the expected information gain, as discussed in the lecture.

To compare the two versions of IMPORTANCE, you should examine each of them by doing the following steps:

1. Learn a decision tree from the data in `training.txt`
2. Document the tree you got in your report
3. Classify all examples in the test-set (given in the text-file `test.txt`), and calculate the accuracy of the learner by comparing to the correct classification of the examples in the test-set.

**Discuss your findings:**

- What can you conclude about the results you have obtained? Which IMPORTANCE is better, and why?
- Do you get the same result if you run the random IMPORTANCE several times?
- What happens when you run the learner based on Information Gain several times?

## Experiment Data

Two data-files are in the ZIP-file you have downloaded. The two data files have the same format: Each line describes an object, the first seven numbers are the attributes, the last number is the class of that object. All attributes as well as the class take values `1` or `2`, and you can take advantage of this to simplify your code if you want.

**Example:**
The first line in the training data is:

`1 1 2 2 1 1 1 1`.

This means that for this object, we have

- Attribute 1 = `1`,
- Attribute 2 = `1`,
- Attribute 3 = `2`,
- Attribute 4 = `2`,
- Attribute 5 = `1`,
- Attribute 6 = `1`,
- Attribute 7 = `1`,
- ... and that the object comes from class `1`.

The test data has the same format. The class label for the test data is not to be used by the decision tree during learning or classification, only to quantify the learning algorithm's accuracy afterwards.