Exercises for
Pattern Analysis
Marco Bögel, Sebastian Käppler
Assignment 4, 19.05.2015

General Information:

Lecture (3 SWS): Mo 08.30 – 10.00 (H16) and Tue 08.15 – 09.45 (H16)
Exercises (1 SWS): Tue 12.15 – 13.15 (02.134-113) and Thu 8.30 – 9.30 (E1.12)
Certificate: Oral exam at the end of the semester
Contact: marco.boegel@fau.de
sebastian.kaeppler@fau.de

CARTs – Classification

Exercise 1 Blood pressure can be categorized as follows:

Category	**Systolic**		**Diastolic**
Normal	< 120	AND	< 80
Prehypertension	120 – 139	OR	80 – 89
High blood pressure			
Stage 1	140 – 159	OR	90–99
Stage 2	> 160	OR	> 100

(Information was obtained from *National Institues of Health.*)

Design a binary classification tree for the four blood pressure categories.

(a) Draw the hierarchical tree structure.

(b) Draw the partitions of the tree in 2-D feature space.

Exercise 2 In a classification tree, an objective function I_j for a node j with a feature set $\boldsymbol{S}_j$ is used for the process of tree creation. This exercise examines the information gain function:

(a) Write down the information-gain function I_j for a node j.

(b) Write down the entropy function $H(\boldsymbol{S}_j)$ for a node j and for N classes.

(c) Show that the entropy function is a concave function. What is the benefit of this property with respect to numerical optimization methods?

(d) Assume a classification problem with $N = 2$ classes is given. Simplify the general form of $H(\boldsymbol{S}_j)$ which is one-dimensional in this case.

(e) For the case of $N = 2$ classes: Calculate the minimum and the maximum values of $H(\boldsymbol{S}_j)$ and draw the function in the interval $[0; 1]$

(f) How does the concavity of the entropy function help to solve the overall optimization problem for the classification tree?

Exercise 3 **Matlab exercise**

The goal of this exercise is to implement the **information gain** as impurity measurement for a CART node.

(a) Download the Matlab code from the exercise homepage.

(b) *CART_example.m* generates some sample data for two classes and uses *train_cart_leavenode.m* to find a split for the data using the impurity measurement implemented in *information_gain.m*.

(c) You only need to implement the information gain in *information_gain.m*. Everything else does not need to be changed.

(d) Think about why the maximum of the information gain can not be obtained using an optimizer like gradient ascent.