Prof. Dr. M. Grohe
E. Fluck, N. Runde

Exercise Sheet 1

Due date: Monday, April 24 until 13:00

- Please upload your solutions to Moodle. The upload will be possible from Thursday, April 20 12:00.
- The upload is possible until **Monday, April 24 13:00**.
- Hand in your solutions in groups of **two to three students**. You can find exercise partners using the Exercise Group forum.
- You need to **join a group** in Moodle via the "Submission Group Assignment" until **Thursday, April 20 11:00**.
- Hand in the solutions of your group as a single PDF file.
- The solutions for this exercise sheet will be published on **Monday, April 24 13:00**.
- A discussion regarding this exercise sheet will take place on **Friday, April 28 14:30** in room AH II.

Exercise 1 (Nearest Neighbour Classification) — 4 points

We consider the k-nearest neighbour classification algorithm in 3-dimensions. Given is a training set consisting of 20 examples together with their classification (1 or -1), and a list of 5 query points. These can be found in the file `nn.py` that has been uploaded along with this sheet.

Classify the 5 query points using the k-nearest neighbour algorithm, for each of the following four configurations:

a) $k = 2$ with Manhattan distance,

b) $k = 3$ with Manhattan distance.

c) $k = 2$ with Euclidean distance,

d) $k = 3$ with Euclidean distance,

When ties occur, indicate them with class label „0".

Hint: Solve this by writing a program that does the job for you.

- Give the results of your classifications in form of a table.
- You do not need to worry about the precision of representations of real numbers.
- You do not need to turn in your code (code submissions will be ignored, only the answers count).

Exercise 2 (Decision Trees) 2 + 2 + 2 = 6 points

Recall that a propositional formula is built from Boolean variables $X_1, \ldots, X_n$ using $\neg$, $\wedge$ and $\vee$. For all variables X, the formulas X and $\neg X$ are called *(positive / negative) literals*. Every propositional formula with n variables represents a Boolean function from $\{0,1\}^n$ to $\{0,1\}$, via the usual notion of (satisfying or non-satisfying) assignments.

A propositional formula is in *k-CNF*, if it is a conjunction of disjunctions of at most k literals, that is, of the shape $\bigwedge_{i=1}^{m} \bigvee_{j=1}^{m_i} L_{i,j}$ with $m_i \leq k$ for all $i = 1, \ldots, m$.

A propositional formula is in *k-DNF*, if it is a disjunction of conjunctions of at most k literals, that is, of the shape $\bigvee_{i=1}^{m} \bigwedge_{j=1}^{m_i} L_{i,j}$ with $m_i \leq k$ for all $i = 1, \ldots, m$.

Answer the following tasks.

a) Give an example of a 2-CNF over 3 variables X_1, X_2, X_3 that has no 2-DNF representation. Explain why it has no 2-DNF representation.

b) Now give an example of a 2-DNF over X_1, X_2, X_3 that has no 2-CNF representation. Explain why it has no 2-CNF representation.

c) Prove the following statement: If f is a Boolean function that can be represented by a decision tree of height $k \in \mathbb{N}$, then f can be represented by both a k-CNF and a k-DNF.

Exercise 3 (Perceptron) **1+1+1+2=5 points**

We want to use the Perceptron algorithm for linear classification on the instance space $\{-1,1\}^n$ where $n \in \mathbb{N}$ is an odd number. The target function (unknown to the algorithm) is the *majority function* $\mathrm{maj}\colon \{-1,1\}^n \to \{-1,1\}$ with

$$\mathrm{maj}(\mathbf{x}) = \begin{cases} 1 & \text{if more than } n/2 \text{ of } \mathbf{x}\text{'s entries are positive} \\ -1 & \text{otherwise.} \end{cases} \qquad (*)$$

We consider a training sequence $S = \big((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_k, y_k)\big)$ of $k \in \mathbb{N}$ data items $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \{-1,1\}^n$ together with their class labels $y_1, \ldots, y_k \in \{-1,1\}$.

a) Show that $\mathrm{maj}\colon \{-1,1\}^n \to \{-1,1\}$ is realisable by a homogenous linear separator by specifying a suitable weight vector $\widehat{\mathbf{w}}$ satisfying $\mathrm{maj}(\mathbf{x}) = \mathrm{sgn}\big(\langle \widehat{\mathbf{w}}, \mathbf{x}\rangle\big)$ for all $\mathbf{x} \in \{-1,1\}^n$.

b) Using the weight vector from part a), derive an upper bound on the number of weight vector updates $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ the Perceptron algorithm performs when run on S.

c) Find the smallest possible number of updates $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ after which the Perceptron algorithm terminates. For all $k \in \mathbb{N}$, $k \geq 1$, describe a training sequence S (with k examples) that for which the algorithm achieves this lower bound, and argue that it does.

d) If the domain of the target function is extended to $\mathbb{R}^n$ (leaving the definition $(*)$ unchanged), can we still find a consistent linear separator for any given training sequence S?

Exercise 4 (k-Means Algorithm) **2+1+2=5 points**

Consider the following set of points in $\mathbb{R}^2$ and the execution of the 3-Means Algorithm on it.

$$S = \{A = (2, 12), B = (3, 11), C = (3, 8), D = (5, 4),$$
$$E = (7, 5), F = (7, 3), G = (10, 8), H = (13, 8)\}$$

a) Give all intermediate clusters and their centers during the execution of the 3-means algorithm on S with the initial cluster means $z^1 = A$, $z^2 = B$, $z^3 = C$.

b) Draw a coordinate system and, in it, indicate (1) the points of S; (2) the three final cluster means; and (3) the three final cluster regions (i. e. the set of points which are closer to the corresponding centroid than to any of the other two).

Describe the lines that separate the regions algebraically as linear equations. Justify the correctness of your expressions.

c) Find a different initialisation of the centroids (as opposed to part (a)) for which the execution of the 3-means algorithm yields a different set of final clusters. Justify your solution.