Prof. Dr. M. Grohe E. Fluck, P. Lindner

Exercise Sheet 1

Due date: Monday, May 4 until 15:00

- Please upload your solutions to Moodle.
- Hand in your solutions in groups of **two to three students**. You can find exercise partners using the Exercise Group forum.
- Please mail the names of you and your group partners to fluck@cs.rwth-aachen.de until **Monday, April 27 15:00**. We will then model the groups in Moodle. Only after that will you be able to upload solutions.
- Please hand the solutions of your group in as a single PDF file.
- You will not be able to change your upload.
- The topics needed for exercise 4 will be published in week 2.
- The solutions for this exercise sheet will be published on **Monday, May 04 15:00**.
- A discussion regarding this exercise sheet will take place on **Friday, May 08 13:00** via Zoom.

Exercise 1 (Nearest Neighbor Classification) — 5 points

Consider the k-nearest neighbor classification algorithm. We consider the variants with $k = 2$ and $k = 3$ here as well as the Euclidean ($\|x - y\|_2$, using the L2-norm) and the Manhattan distance ($\|x - y\|_1$, using the L1-norm).
Let the training set S contain the 20 points listed in the file `data-1.txt`. Your task is to fill a table for the four variants of the k-nearest neighbor algorithm for the following datapoints:

$$(4, 3, 3),\ (4, -1, 1),\ (-2, 4, 5),\ (-2, -6, -1),\ (6, 0, 2)$$

Exercise 2 (Decision Trees) — 4 points

Let $f(x, y, z) = (x \vee z) \wedge (z \vee y)$ and assume your training set S to uniquely specifies this function. Graphically construct a decision tree over the Boolean-valued $\{0, 1\}$ feature set $\{x, y, z\}$ using the algorithm from the lecture. Make sure that your tree has as few nodes as possible. For every split in the decision tree, give an argument to support your choice of the discriminating feature.

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

Let our instance space be $X = \{-1, 1\}^n$ where n is an odd number and consider the target function $\mathrm{maj} : X \to \{-1, +1\}$. It is valued $+1$ when a *majority* (at least half) of the input vector's values are positive, and -1 otherwise (since n is odd, there is always a majority). In this exercise we want to realize this target function with the Perceptron algorithm.

a) Assume the size of the training set S is k. What is the minimal number of updates $\mathbf{w} \leftarrow \mathbf{w} + y\mathbf{x}$ that the Perceptron needs until the function $\mathbf{x} \to \mathrm{sgn}(\mathbf{w} \cdot \mathbf{x})$ is consistent with S?

b) Specify a normalized vector $\mathbf{w}$ such that $\mathrm{maj}(\mathbf{x})$ and the homogeneous linear separator $\mathrm{sgn}(\mathbf{w} \cdot \mathbf{x})$ compute the same function. Using your normalized vector $\mathbf{w}$ find an upper bound on the number of updates of $\mathbf{w}$ the Perceptron Algorithm performs for any training set for the function maj.

c) If we change X to $\mathbb{R}^n$, can we still find a linear separator that realizes maj?

Justify your answers.

Exercise 4 (k-Means Algorithm) **2+1+2+1=6 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 of the 3-Means Algorithm on S with the initial cluster means A, B, C.

b) Draw a coordinate system and mark the points of S as well as the final 3 cluster means in it. Also draw the regions of the final cluster means, i.e., draw for each final cluster mean the region of points which are closer to this mean than to any of the two other final means. Describe the lines that separate the regions algebraically via function terms. Justify the correctness of your algebraic descriptions.

c) Find a different set of initial cluster means (chosen from the data points) for which the execution of the 3-Means Algorithm on S yields a different set of final clusters. Justify your solution.

d) Can you find k initial cluster centers for some k such that the k-Means Algorithm returns $\{A, G\}$ as a cluster? If yes, which ones? If not, why not?