Lab Class ML:II

Until **Monday, Nov. 25th, 2019, 11:00 am CET**, the following exercises must be submitted:

Students receiving 4.5 ECTS: 2, and 3.
Students receiving 6 ECTS: 1, 2, and 3.

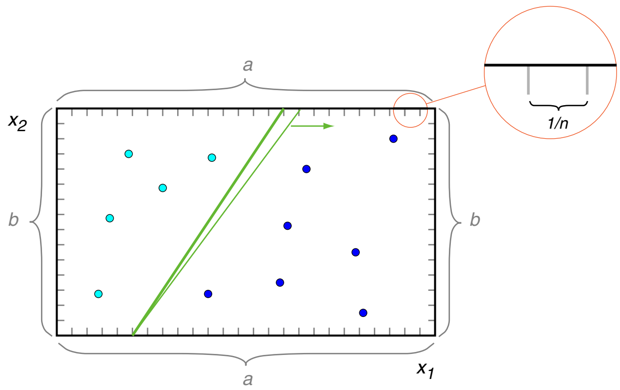
Lab Class General Instructions. Exercises should be completed in groups of three students; which and how many exercises you must submit depends on the number of ECTS credits awarded for the course, based on which degree programme you're studying:

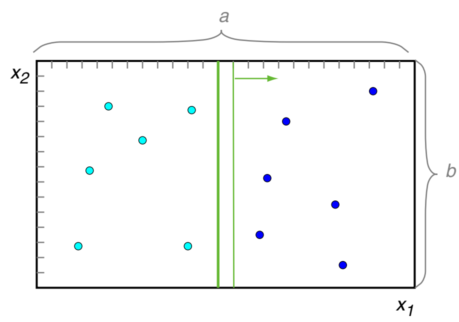
	4.5 ECTS	**6 ECTS**
Degree programmes	CS4DM, CSM, MI, HCI *(enrolled 2018 or earlier)*	DE, HCI *(enrolled 2019 or later)*
For admission to the final exam:		
Must reach at least	80 points	120 points
Out of maximum	120 points	180 points

For each exercise sheet, all required exercises must be submitted until the stated deadline. Each individual exercise will receive a score between zero and ten. Submissions received after the deadline will not be considered for grading. In order to be admitted to the final exams, you must reach the minimum total score shown.
Upload your solutions to Moodle as a single `.pdf` file, plus one `.ipynb` file for any programming exercises. Make it clear which solution corresponds to which exercise. Programming exercises are marked with $\boxed{\text{P}}$ – their solutions must be documented extensively with docstrings and/or inline comments. ml.weimar.webis.de

Exercise 1 : Hypothesis Space

Let x_1 and x_2 be two attributes for the description of objects; the sizes of their domains are $|X_1|$ and $|X_2|$. An object can belong to class 0 or class 1, illustrated as light and dark dots in the figure. Let the hypothesis space H comprise all discrimination lines that can be drawn in the two-dimensional feature space X.





(a) Compute an upper bound for $|H|$, if arbitrary linear hypotheses are allowed (see left figure).

(b) Compute an upper bound for $|H|$, if arbitrary linear hypotheses like $x_1 - \beta = 0, \beta \in X_1$ are allowed (see right figure).

Exercise 2 : Concept Learning

Given is the following training set D, which you have obtained as co-driver by observing your friend:

	Weekday	Mother-in-the-car	Mood	Time of day	run-a-red-light
1	Monday	no	easygoing	evening	yes
2	Monday	no	annoyed	evening	no
3	Saturday	yes	easygoing	lunchtime	no
4	Monday	no	easygoing	morning	yes

Let the set H contain hypotheses that are built from a conjunction of restrictions for attribute-value combinations; e. g. $\langle \textit{Monday}, \textit{yes}, ?, ? \rangle$.

(a) Apply the Find-S algorithm for the example sequence 1, 2, 3, 4.

(b) Apply the Candidate-Elimination algorithm for the example sequence 1, 2, 3, 4, and identify the boundary sets S and G.

(c) What is the version space $V_{H,D}$ for this example? Can a version space contain hypotheses that are neither in the corresponding S nor in G? If so, how?

Exercise 3 : $\boxed{\text{P}}$ Concept Learning

Develop a simple Python implementation of the Candidate-Elimination algorithm for the EnjoySport learning task discussed in the lecture.

(a) Hypotheses and examples will be represented as tuples of strings. Use the following functions to generate the minimally and maximally specific hypotheses:

```
def g_0(n):
    return ("?",) * n
def s_0(n):
    return ("⊥",) * n # NOTE: feel free to use a different symbol
                      # in case of encoding issues
```

(b) Implement a function `more_general(h1, h2)` that returns `True` when $h_1 \geq_g h_2$ holds. Example:

```
>>> more_general(("sunny", "warm", "?"), ("sunny", "?", "normal"))
False
```

(c) Implement a function `min_generalizations(h, x)` that returns the minimal generalizations of the given hypothesis `h`, such that the new hypotheses are fulfilled by `x`.

Example:

```
>>> min_generalizations(h=("sunny", "warm"), x=("rainy", "warm"))
[("?", "warm")]
```

(d) Implement a function `min_specializations(h, domains, x)` for a hypothesis `h` and an example `x`. The argument `domains` is a list of lists, in which the i-th sub-list contains the possible values of feature i. The function should return all minimal specializations of hypothesis `h` with respect to `domains`, such that the returned hypotheses are not fulfilled by `x`. Example:

```
min_specializations(h=("?", "x"),
                    domains=[["a", "b", "c"], ["x", "y"]],
                    x=("b", "x"))
[("a", "x"), ("c", "x"), ("?", "⊥")]
```

(e) Implement the Candidate-Elimination algorithm from the slides as a Python function with the following signature: `candidate_elimination(examples)`, where `examples` is a list of $(n+1)$-tuples. The first n elements of each tuple should be the feature values, and the last element is the value of the target concept (you may assume that this value is always either `True` or `False`). Your function should return a 2-tuple containing the sets G and S. Hint: Use the previously implemented functions where appropriate. Start by computing the domains of the features, and generating g_0 and s_0 of the correct dimension.

(f) Given is the following set of examples for the target concept EnjoySport:

	Sky	Temp	Humid	Wind	Water	Forecast	Sleep	EnjoySport
1	sunny	warm	normal	strong	warm	same	long	True
2	sunny	warm	high	strong	warm	same	long	True
3	rainy	cold	high	weak	warm	change	medium	False
4	sunny	warm	high	strong	cool	change	long	True
5	cloudy	cold	high	strong	cool	change	short	False

Run your implementation on the examples in the order given above, and report the values of G and S that you obtain.

(g) Produce a visualization of the hypothesis space $V_{H,D}$ spanned by G and S.

Hint: printing out the hypotheses in $V_{H,D}$ in a sensible arrangement is sufficient.