Last Name: First Name:

Matriculation Number:

Exam
Artificial Intelligence 2

2023-10-10

	To be used for grading, do not write here											
prob.	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	5.1	5.2	Sum	grade
total	7	11	7	10	11	11	9	9	6	6	87	
reached												

Organizational Information

Please read the following directions carefully and acknowledge them with your signature.

1. Please place your student ID card and a photo ID on the table for checking.
2. You can reach 87 points if you fully solve all problems. You will only need 80 points for a perfect score, i.e. 7 points are bonus points.
3. No resources or tools are allowed except for a pen.
4. In particular, you are not allowed any electronic devices – phones, smart watches, earbuds, smart rings, … on your person. Put them into your backpack or your pockets.
5. You have 90 min (sharp) for the exam.
6. Write the solutions directly on the sheets, no other paper will be graded.
7. If you have to abort the exam for health reasons, your inability to sit the exam must be certified by an examination at the University Hospital. Please notify the exam proctors and have them give you the respective form.
8. Please make sure that your copy of the exam is complete (12 pages excluding cover sheet and organizational information pages) and has a clear print. **Do not forget to add your personal information on the cover sheet and to sign this declaration.**

Declaration: With my signature I certify having received the full exam document and having read the organizational information above.

Erlangen, 2023-10-10 ....
(signature)

Please consider the following guidelines to avoid losing points:

- If you continue an answer on another page, clearly give the problem number on the new page and a page reference on the old page.
- You can always ask for the translation or explanation of a non-technical word.
- If you do not want something to be graded, clearly cross it out. Adding a wrong statement to a correct solution may lead to deductions.
- The instructions "Give X", "List X" or similar mean that only X is needed. If you additionally justify your answer, we may or may not give you partial credit.
- The instruction "Assume X" means that X is information that your answer may use.
- The instruction "Model X as a Y" means that you have to describe X formally and exactly as an instance of Y using the definition of Y from the lecture.
- If you are uncertain how long or complex an answer should be, use the number of points as an indication: 1 point roughly corresponds to 1 minute.
- In all calculation questions, you have to simplify as much as reasonably possible without a calculator. For example, $\log 2$ or 3^7 should not be calculated, but $0.4 \cdot 0.3 \cdot 0.5 = 0.06$ should be.

1 Probabilities

Problem 1.1 (Python)

1. Consider the Python *program* below. 4 pt

```
# input: two lists of real numbers between 0 and 1
def bar(x, y):
l = len(x)

    m = len(y)
    res = []
    for i in range(l):
        row = []
        for j in range(m):
           row.append(x[i]*y[j])
        res.append(row)
    return res
```

Assuming the *inputs* represent *probability distributions*, which probability-related operation does the function bar compute? State any assumptions that need to be made about the *inputs*.

2. Assume *random variables* X with *domain* $\{0, \dots, m-1\}$ and Y with *domain* $\{0, \dots, n-1\}$. Assume the Python object C holds their *joint probability distribution* $P(X, Y)$, i.e., $C[i][j] = P(X = i, Y = j)$. 3 pt
Complete the definition of E in the *program* below in such a way that it holds the *probability distribution* $P(X)$, i.e., $E[i] = P(X = i)$.

```
# m, n, C are defined as described above

E =
```

Problem 1.2 (Calculations)
Assume *random variables* X, Y both with *domain* $\{0, 1, 2\}$, whose *joint probability distribution* $P(X, Y)$ is given by

x	y	$P(X = x, Y = y)$
0	0	a
0	1	b
0	2	c
1	0	d
1	1	e
1	2	f
2	0	g
2	1	h
2	2	i

1. Give all *subsets* of the *probabilities* $\{a, b, c, d, e, f, g, h, i\}$ that *sum* to 1. 2 pt

2. In terms of $a, b, c, d, e, f, g, h, i$, give $P(X \neq 0)$. 2 pt

3. In terms of $a, b, c, d, e, f, g, h, i$, give $P(X = 1, Y = 0)$. 1 pt

4. In terms of $a, b, c, d, e, f, g, h, i$, give $P(X = 1 \mid Y = 0)$. 2 pt

5. In terms of $a, b, c, d, e, f, g, h, i$, give $P(X + Y = 2)$. 2 pt

6. In terms of $a, b, c, d, e, f, g, h, i$, give $P(X + Y = 2 \mid X > Y)$. 2 pt

2 Bayesian Reasoning

Problem 2.1 (Bayesian Calculations)
Consider a disease with a prevalence of 1/1000, i.e., 1 in 1000 people have it. You are using a test that gives a yes/no answer for whether a person has the disease. However, the test randomly returns the wrong result 1% of the time.

1. Model this situation using *random variables*. State all *probabilities* whose *values* are given in the text. 3 pt

2. Calculate the *probability* of a test returning yes. 2 pt

3. You are using the test on a person, and it returns yes. Calculate the *probability* that she has the disease. 2 pt

Problem 2.2 (Bayesian Networks)
Consider the following situation about a *chess* game:

- The *outcome* O can be a win for white (w), a win for black (b), or a draw (d).
- The *players* have experience levels E_w and E_b, whose possible values are fresh (f), experienced (e), and professional (p), and that allow making predictions about the result of a game.
- You have placed a bet on the *outcome* (without knowing the *players*), and the *outcome* will determine if you gain (g) or lose (l) money (M).

You want to model this situation as a *Bayesian network*.

1. Give the set of *random variables* and their *domains*. 2 pt

2. Give a good *variable ordering* and draw the resulting *Bayesian network*. 2 pt

3. You have already placed the bet. What does that mean for the relationship between O and M? How does that affect the memory needed for the *conditional probability tables* of the *network*? 2 pt

4. Assume your *network* is $E_w \to O \leftarrow E_b \to M$ (which may or may not be correct). How many entries does the *conditional probability table* for O have? 2 pt

5. Assume again your *network* is $E_w \to O \leftarrow E_b \to M$. Give the *formula* for $P(O \mid M, E_b)$ in terms of the entries of the *probability tables* of the *network*. 2 pt

3 Markovian Reasoning

Problem 3.1 (Hidden Markov Models)
Consider the following situation:

- We investigate the rainfall at a certain location. Each year the rainfall is high (r_1), medium (r_2), or low (r_3).
- We know this causes a groundwater condition, which is either strong (g_1) or weak (g_2).

We have modeled this situation as a *stationary* and first-*order hidden Markov model* with two families of *random variables* R_a (rainfall) and G_a (groundwater), each indexed by year number a.

Transition Model *Sensor Model*

$$T = \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.6 \\ 0 & 0.1 & 0.9 \end{pmatrix} \qquad S = \begin{pmatrix} 0.4 & 0.6 \\ 0.2 & 0.8 \\ 0.25 & 0.75 \end{pmatrix}$$

1. Give the *state* and *evidence variables* and their *domains*. 2 pt

2. Which *probabilities* are captured by the entries $\mathbf{T}_{ij}$ and $\mathbf{O}_{t_{ij}}$? 2 pt

3. The rainfall was high last year and is low this year. Give the *probability distribution* of this year's groundwater condition. 2 pt

4. What is the purpose of the *smoothing algorithm*? 2 pt

5. Given *evidence variables* $G_1 = e_1, \ldots, G_a = e_a$, the *smoothing algorithm* can be written in *matrix* form as $P(R_k \mid e_{1:a}) = \alpha \mathbf{f}_{1:k} \mathbf{b}_{k+1:a}$. Give the recursive equations for $\mathbf{f}$ and $\mathbf{b}$ and explain the *values* of the *matrices* O. 3 pt

Problem 3.2 (Decision Processes and Utility)
Consider an *agent* moving along a circular arrangement of 8 locations as indicated below.

$$\begin{array}{ccccccc} 0 & \rightarrow & 1 & \rightarrow & 2 & \rightarrow & 3 \\ \uparrow & & & & & & \downarrow \\ 7 & \leftarrow & 6 & \leftarrow & 5 & \leftarrow & 4 \end{array}$$

The *agent*'s movement is as follows:

- It can move −2, −1, 0, 1, or 2 steps (negative numbers represent backwards movement).
- The double steps result in moving 2 locations with probability 60%, and 1 location in the opposite direction otherwise.
- The single steps result in moving 1 location with probability 90%, and no move otherwise.
- The zero step results in no move.

The *agent*'s *goal* is to move to location 7.

1. Model this situation as a *Markov Decision Process* $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, s_0, R \rangle$. Use a *reward function* that uses a constant *reward* for *non-goal states*. 4 pt

2. State the *Bellman equation* for $\gamma = 0.5$. Then using initial *utilities* $U(s) = 0$ for all *states*, compute the *value* of $U(4)$ after two *value iteration* steps. 3 pt

3. Give an *optimal policy* π^*. 2 pt

4. Now assume we use a *partially observable Markov decision process* because the *agent* is unable to tell what move an *action* resulted in. Assume we know the *agent* is initially in location 4. Give the *belief state* after a double step forward. 2 pt

4 Learning

Problem 4.1 (Decision Trees and Lists)
Consider an unknown *word* $W \in \{I, you, he, she, it, we, they\}$. You are allowed to ask the following questions about W:

A length of the word; possible answers: 1, 2, 3, 4

B occurrence of the letter h; possible answers: yes/no

C last letter of the word; possible answers: I, u, e, t, y

D first letter of the word; possible answers: I, y, h, s, i, w, t

1. Draw the *decision tree* for W that arises from asking the questions in the order given above. (Do not ask additional questions if the word can already be identified.) 3 pt

2. Which choice would the *information gain algorithm* make first? Justify your answer. 2 pt

3. Give the size of the smallest *decision list* (measured as the sum of the numbers of *literals* in all tests) if tests may use arbitrarily many *literals* of the form $Question = Answer$. 2 pt

4. Give all minimal sets $Q \subseteq \{A, B, C, D\}$ of questions for which the *determination* $Q \succ W$ holds? 2 pt

Problem 4.2 (Support Vector Machines)
Consider the following dataset of points $\mathbf{x}_1, \mathbf{x}_2 = \langle \mathbf{x}_1, \mathbf{x}_2 \rangle$ in $\mathbb{R}^2$ that are classified as either $y = +1$ or $y = -1$:

$\mathbf{x}_1$	$\mathbf{x}_2$	y
2	3	1
−2	−2	−1
4	−3	−1

1. Give the *hypothesis space* for finding a *linear separator*. 2 pt

2. Give a *linear separator* $h(\mathbf{x}_1, \mathbf{x}_2)$ for the dataset. 3 pt

3. Transform the dataset into a 1-dimensional dataset using the transformation $T(\mathbf{x}_1, \mathbf{x}_2) = {\mathbf{x}_1}^2 + {\mathbf{x}_2}^2$. 2 pt

4. What does it mean, intuitively, if a *linear separator* exists for a dataset after this transformation? 2 pt

5 Natural Language Processing

Problem 5.1 (Part-of-Speech Tagging)

1. Briefly explain what *part-of-speech tagging* means. 2 pt

2. What is the role of the width of the *prediction window* when *machine learning POS tags*? 2 pt

3. Explain (in about 2 sentences) the role of *word embedding* when *learning POS tags*, and the idea behind tfidf. 2 pt

Problem 5.2 (Grammars)
Consider the following *probabilistic grammar*:

$$\begin{array}{lcl} S & \to & NP\ VP[1] \\ NP & \to & Article\ Noun[0.6] \mid Name[0.4] \\ VP & \to & Verb[0.5] \mid TransVerb\ NP[0.5] \\ Article & \to & \text{the}[0.7] \mid \text{a}[0.3] \\ Noun & \to & \text{stench}[0.2] \mid \text{breeze}[0.3] \mid \text{wumpus}[0.5] \\ Name & \to & \text{John}[0.3] \mid \text{Mary}[0.7] \\ Verb & \to & \text{smells}[1] \\ TransVerb & \to & \text{sees}[0.6] \mid \text{shoots}[0.4] \end{array}$$

1. Using this *grammar* as an example, explain the difference between *structurals rule* and *lexicon*. 2 pt

2. Give the *probability* of the *sentence* 2 pt

   John sees the wumpus

   (You have to give the *expression* with concrete *values* plugged in, but you do not have to *compute* the result.)

3. Now assume we do not know the *probabilities* of the *production*, and our *corpus* is 2 pt

   John sees the wumpus. The wumpus smells. John shoots the wumpus.

   Give the *probability* that we can learn for $NP \to Name$ from this *corpus*.

This page was intentionally left blank for extra space

This page was intentionally left blank for extra space