Last Name: First Name:

Matriculation Number:

Exam
Artificial Intelligence 2

October 10, 2023

Please ignore the QR codes; do not write on them, they are for grading support

	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	8	8	10	11	11	9	9	6	6	85	
reached												

The “solutions” to the exam/assignment problems in this document are supplied to give students a starting point for answering questions. While we are striving for helpful “solutions”, they can be incomplete and can even contain errors even after our best efforts.
In any case, grading student’s answers is not a process of simply “comparing with the reference solution”, therefore errors in the “solutions” are not a problem in this case. If you find “solutions” you do not understand or you find incorrect, discuss this on the course forum and/or with your TA and/notify the instructors. We will – if needed – correct them ASAP.

In the course Artificial Intelligence I/II we award bonus points for the first student who reports a factual error in an old exam. (Please report spelling/formatting errors as well.)

1 Probabilities

Problem 1.1 (Python)

1. Consider the Python program below. 4 Points

```
# 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.

Solution: The joint probability distribution $P(X = i, Y = j) = P(X = i) \cdot P(Y = j)$ of two stochastically independent variables with distributions $x[i] = P(X = i)$ and $y[j] = P(Y = j)$.

---

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 Points
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 =
```

Solution: `E = [sum(C[i]) for i in range(m)]`.

Problem 1.2 (Calculations)
Assume random variables X, Y both with domain $\{0, 1, 2\}$, whose joint 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 Points

   Solution: Only $\{a, b, c, d, e, f, g, h, i\}$

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

   Solution: $d + e + f + g + h + i$

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

   Solution: $c + e + g$

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

   Solution: $g/(d + g + h)$

2 Bayesian Reasoning

Problem 2.1 (Bayesian Calculations)
Assume you are trying to relate economic development and your business results. You have collected the following data:

- The economy does well 40% of the time and badly otherwise.
- Your business does well 30% of the time and badly otherwise.
- If your business does well, the economy did well 80% of the time.

You model the problem using two Boolean random variables E (economy does well) and B (business does well). You also abbreviate the events $E = true$ and $B = true$ as e and b.

1. By filling in the gaps below, state for each number in the text above, which probability it describes. 2 Points

   1. $P($_____$) = 0.4$

   2. $P($_____$) = 0.3$

3. $P($_____$) = 0.8$

Solution: $P(e) = 0.4$, $P(b) = 0.3$, and $P(e|b) = 0.8$

2. Using Bayes' Rule, compute the probability that your business does well if the economy does. 2 Points

Solution: $P(b|e) = P(e|b) \cdot P(b)/P(e) = 0.8 \cdot 0.3/0.4 = 0.6$

3. Explain how we can compute all values in the joint distribution of E and B. 4 Points
(You can omit purely mathematical computations unrelated to probabilities if you mention what they do.)

Solution: The distribution contains 4 unknown values: $P(b,e)$, $P(b,\neg e)$, $P(\neg b,e)$ and $P(\neg b,\neg e)$.
We know 4 properties about them:
- $P(e) = P(b,e) + P(\neg b,e) = 0.4$
- $P(b) = P(b,e) + P(b,\neg e) = 0.3$
- $P(e|b) = P(b,e)/P(b) = 0.8$
- $P(b,e) + (b,\neg e) + P(\neg b,e) + P(\neg b,\neg e) = 1$

From those we can compute the 4 values.
The result (not needed for full points) is

	e	$\neg e$	Σ
b	0.24	0.06	0.3
$\neg b$	0.16	0.54	0.7
Σ	0.4	0.6	1

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 Points

Solution: Variables O with domain $\{w,b,d\}$, E_w and E_b both with domain $\{f,e,p\}$, and M with domain $\{g,l\}$.

2. Give a good variable order and draw the resulting Bayesian network. 2 Points

Solution: Order: E_wE_bOM or E_bE_wOM. Network: $E_w \to O \leftarrow E_b$ and $O \to M$.

3. 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 Points

Solution: The entries are $P(O = z|E_w = x, E_b = y)$ where x, y, z range over the respective domains. So there are $3 \cdot 3 \cdot 3 = 27$ entries. Exploiting redundancies, we need to store only 18.

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

Solution: $P(O|M, E_b) = P(O|E_b)$ and $P(O = z|E_b = x) = \Sigma_{y\in\{f,e,p\}} P(O = z|E_b = x, E_w = y)$

5. 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 Points

Solution: The edge $O \to M$ is deterministic. We do not need to store a probability table for M; instead we have to store the function that computes the value of M from the value of O.

3 Markovian Reasoning

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

- We make annual observations about 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

$$T = \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.1 & 0.3 & 0.6 \\ 0 & 0.1 & 0.9 \end{pmatrix}$$

Sensor Model

$$S = \begin{pmatrix} 0.4 & 0.6 \\ 0.2 & 0.8 \\ 0.25 & 0.75 \end{pmatrix}$$

1. Give the state and evidence variables and their domains. 2 Points

Solution: Evidence variables $G_a \in \{g_1, g_2\}$, state variables $R_a \in \{r_1, r_2, r_3\}$.

2. Which probabilities are captured by the entries T_{ij} and S_{ij}? 2 Points

Solution: $T_{ij} = P(R_{a+1} = r_j|R_a = r_i)$ and $S_{ij} = P(G_a = g_j|R_a = r_i)$.

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

Solution: $P(G_a|R_a = r_3) = (0.25, 0.75)$ (Last year's rainfall is irrelevant because the sensor model is first-order.)

4. What is the purpose of the smoothing algorithm? 2 Points

Solution: To estimate past states based on observations of all the evidence (even after the state in question).

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

Solution: $f_{1:k+1} = \alpha O_{k+1} T^t f_{1:k}$ and $b_{k+1:a} = T O_{k+1} b_{k+2:a}$
O_i is a diagonal matrix obtained from the column of S corresponding to e_i.

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 $\langle S, A, P, R\rangle$. Use a reward function that uses a constant reward for non-goal states. 4 Points

Solution: One possible model is

- $S = \{0, \ldots, 7\}$
- $A(s) = \{2, 1, 0, -1, -2\}$
- $P(s'|a, s)$ is given by
  - $a = 0$: $P(s|0, s) = 1$
  - $|a| = 1$: $P(s +_8 a|a, s) = 0.9$, $P(s|a, s) = 0.1$
  - $|a| = 2$: $P(s +_8 a|a, s) = 0.6$, $P(s -_8 a/2|a, s) = 0.4$

  where $+_8, -_8$ are addition/subtractions modulo 8. All other probabilities are 0.
- A typical choice is any function R that is high for the goal and slightly negative for other states. E.g., $R(7) = 1$ and $R(s) = -0.1$ otherwise.

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 Points

Solution: $U(s) = R(s) + 0.5 \max_{a \in A(s)} \Sigma_{s' \in S} U(s') \cdot P(s'|s,a)$
First iteration: $U(s) = R(s)$. ($U(s)$ must be computed for all states before the second iteration can be carried out. But if initial utilities are $U(s) = 0$, this is trivial.)
Second iteration: $U(4) = R(4) + 0.5 \max_{a \in A(4)} \Sigma_{s' \in S} U(s') \cdot P(s'|s,a) = -0.1 - 0.1 \cdot 0.5 \max_{a \in A(4)} \Sigma_{s' \in S} P(s'|4,a)$
$-0.1 - 0.05 \max_{a \in A(4)} 1 = -0.1 - 0.05 \cdot 1 = -0.15$
Different choices of reward function in the previous subproblem lead to correspondingly different solutions with $U(s) = 1.5 \cdot r$ where r is the constant negative reward value.

3. Give an optimal policy π^*. 2 Points

Solution: Single steps are faster on average than double steps, except in state 3 where backward movement is as good as forward movement. However, for states $1, 2, 4, 5$, the optimal policy subtly depends on the discount factor and the reward function because a successful double step collects a high reward earlier. Therefore, any step towards the goal was accepted as near-optimal in those states.
Optimal: $\pi^*(7) = 0$, $\pi^*(0) = -1$, $\pi^*(6) = 1$, $\pi^*(3) \in \{-2, 2\}$
Near-optimal: $\pi^*(2), \pi^*(3) \in \{-1, -2\}$, and $\pi^*(4), \pi^*(5) \in \{1, 2\}$

4. Now assume we use a POMDP 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 Points

Solution: The belief state B is a probability distribution over states $s \in S$. The values are $P(6) = 0.6$, $P(3) = 0.4$ and $P(s) = 0$ otherwise.

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, returning from $\{1, 2, 3, 4\}$

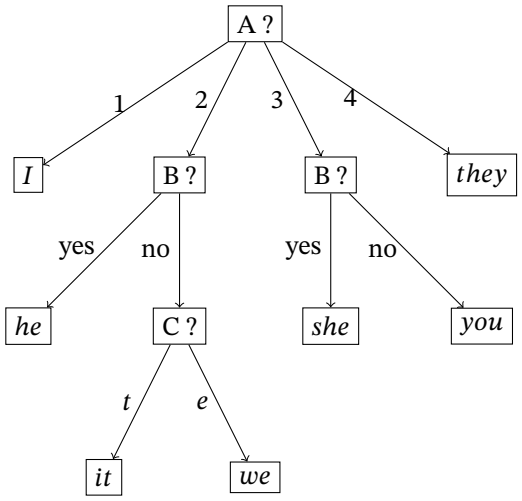
B occurrence of the letter h, returning yes/no

C last letter of the word, returning from $\{I, u, e, t, y\}$

D first letter of the word, returning from $\{I, y, h, s, i, w, t\}$

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 Points

Solution: The tree is



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

Solution: It would ask D because that already identifies the word completely, i.e., maximizes information gain.

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 Points

Solution: 6. It's impossible to be smaller because we need to have 7 end points. And it's straightforward to give a list of that size.

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

Solution: $\{A, B, C\}$ and $\{D\}$

Problem 4.2 (Support Vector Machines)
Consider the following dataset of points $\mathbf{x} = \langle \mathbf{x}_1, \mathbf{x}_2 \rangle$ 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 Points

Solution: The set of functions $\mathbf{w} \cdot \mathbf{x} + b$ for real numbers $\mathbf{w}_1, \mathbf{w}_2, b$. Alternatively, one can use $\mathbb{R}^3$ with some explanation that it holds the tuples $(\mathbf{w}_1, \mathbf{w}_2, b)$.

2. Give a linear separator $h(\mathbf{x})$ for the dataset. 3 Points

Solution: An easy choice is $h(\mathbf{x}) = \mathbf{x}_2$, i.e., the line $x_2 = 0$ separates the points according to y.

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

Solution: The transformed dataset is

$T(\mathbf{x})$	y
13	1
8	−1
25	−1

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

Solution: The two categories are the inside and the outside of a circle around the origin.

5 Natural Language Processing

Problem 5.1 (Part-of-Speech Tagging)

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

Solution: The process of attributing to every word in a corpus its syntactic category, like noun, participle, etc.

2. What is the role of the window width when machine-learning part-of-speech tags? 2 Points

Solution: The size of the context that is kept around the word that is to be tagged. For example, with a window width of 5, the two words before and after are added as input to the learning system.

3. Explain (in about 2 sentences) the role of word embeddings when learning part-of-speech tags, and the idea behind $tfidf$. 2 Points

Solution: A word embedding maps a word to a vector of numbers that can be used as input to a neural network. $tfidf$ is a specific embedding, whose definition uses the frequency of words in the documents of the corpus to map words to numbers.

Problem 5.2 (Grammars)
Consider the following probabilistic grammar:

S	$\rightarrow$	$NP\ VP$[1]
NP	$\rightarrow$	$Article\ Noun$[0.6] | $Name$[0.4]
VP	$\rightarrow$	$Verb$[0.5] | $TransVerb\ NP$[0.5]
$Article$	$\rightarrow$	the[0.7] | a[0.3]
$Noun$	$\rightarrow$	stench[0.2] | breeze[0.3] | wumpus[0.5]
$Name$	$\rightarrow$	John[0.3] | Mary[0.7]
$Verb$	$\rightarrow$	smells[1]
$TransVerb$	$\rightarrow$	sees[0.6] | shoots[0.4]

1. Using this grammar as an example, explain the difference between grammar rules and lexicon. 2 Points

   Solution: Both are productions of the grammar. Grammar rules define the language in general (S to $Article$, above), the lexicon defines the specific identifiers used in a context ($Noun$ to $TransVerb$ above).

2. Give the probability of the sentence 2 Points

   John sees the wumpus

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

   Solution: $0.4 \cdot 0.3 \cdot 0.5 \cdot 0.6 \cdot 0.6 \cdot 0.7 \cdot 0.5 = 0.00756$

3. Now assume we do not know the probabilities of the productions, and our corpus is 2 Points

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

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

   Solution: We count all subtrees of type NP (N) and among those the ones of type $Name$ (n), then we learn the probability n/N. That yields $2/5$.