Last Name: First Name:

Matriculation Number:

Exam
Artificial Intelligence 2

April, 2025

	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	4.3	5.1	5.2	Sum	grade
total	8	10	7	10	11	11	8	8	6	7	4	90	
reached													

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)
Below, the input J represents the joint probability distribution of random variables X with domain $\{0,\dots,m-1\}$ and Y with domain $\{0,\dots,n-1\}$ in such a way that $J[x][y] = P(X = x, Y = y)$. All probabilities are represented as floating point numbers.

1. Complete the function below so that it returns the probability distribution of Y. 4 pt

```
def probY(J):
```

Solution:

```
def probY(J):
  numXs = len(J)
  return sum([J[v][y] for v in range(numXs)])
```

---

2. Consider the Python program below. 3 pt

```
def foo(J):
  numXs = len(J)
  numYs = len(J[0])
  for x in range(numXs):
    for y in range(numYs):
      px = sum([J[x][v] for v in range(numYs)])
      py = sum([J[v][y] for v in range(numXs)])
      if J[x][y] != px*py:
        return False
  return True
```

Which probability-related operation does the function `foo` compute?

Solution: It checks if X and Y are independent.

---

3. When using the function `foo` above, which effect may cause it to return an incorrect result? 1 pt

---

Solution: Rounding errors — we are returning a Boolean based on floating point equality.

Problem 1.2 (Calculations)
Assume random variables X, Y both with domain $\{0, 1, 2\}$. For some outcomes A, the probabilities are known as follows:

A	$P(A)$
$X = 0$	a
$X = 0 \wedge Y = 0$	b
$X \neq 0 \wedge Y = 0$	c
$X \neq 0 \wedge Y \neq 0$	d

1. It is guaranteed that $a, b, c, d \in [0; 1]$. What other properties about the numbers a, b, c, d are guaranteed to hold? 2 pt
   For example, properties might be of the form $a + b = 1$ or $a < b$.

   Solution: $b \leq a$ and $a + c + d = 1$

2. In terms of a, b, c, d, give $P(X = 0 \mid Y = 0)$ or argue why there is not enough information to compute the value. 2 pt

   Solution: $b/(b + c)$

3. In terms of a, b, c, d, give $P(Y \neq 0)$ or argue why there is not enough information to compute the value. 2 pt

   Solution: $a - b + d$ or $1 - (b + c)$

4. In terms of a, b, c, d, give $P(X + Y = 0 \mid X \cdot Y = 0)$ or argue why there is not enough information to compute the value. 2 pt

   Solution: $b/(a + c)$

5. Now assume that X and Y are independent. Show that c can be computed from a and b. 2 pt

   Solution: We have $b + c = P(Y = 0) = b/a$, which yields $c = b/a - b$.

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.
- When your business does well, the economy does 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 pt
   1. $P($ $) = 0.4$
   2. $P($ $) = 0.3$
   3. $P($ $) = 0.8$

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

2. Calculate the probability that your business does well when the economy does badly. 2 pt

   Solution: $P(b \mid \neg e) = P(\neg e \mid b) \cdot P(b)/P(\neg e) = (1 - 0.8) \cdot 0.3/(1 - 0.4) = 0.1$

3. How can you store the joint distribution of E and B in a minimally big table? 3 pt

   Solution: Store any three values out of $P(b, e)$, $P(b, \neg e)$, $P(\neg b, e)$ and $P(\neg b, \neg e)$. That is sufficient to compute the fourth (because they sum to 1). Alternatively, many other sets of three probabilities are sufficient, e.g., the ones given above.

Problem 2.2 (Bayesian Networks)
Consider the following situation about an alarm clock:

- It fails to ring if its batteries are empty.
- It fails to ring if it the volume is muted.
- You might oversleep if your alarm clock fails.
- You might oversleep if you stay up late.

You want to model this situation as a Bayesian network using Boolean random variables.

1. Give an appropriate set of random variables and their meaning and draw the Bayesian network (using an appropriate variable order). 3 pt

Solution: Variables: F (clock fails), E (batteries empty), M (clock on mute), O (oversleep), U (up late).
Network: $E \rightarrow F \leftarrow M$ and $F \rightarrow O \leftarrow U$

2. Give the probability of the clock failing in terms of the entries of the conditional probability tables of your network. 2 pt

Solution: $P(F^+) = \Sigma_{e,m \in \{true,false\}} P(F^+ \mid B = b, M = m) \cdot P(M = m) \cdot P(E = e)$

Now you decide to model the failing of the clock as a deterministic node.

3. Explain (in about 3 sentences) whether that decision is justified by the description, and how it changes the conditional probability tables of your model. 3 pt

Solution: In that case, $F = M \vee E$, and we do not have to store a CPT for F and only need to store that definition.
It is not justified. The description does imply that F holds if M or E does. But the description does not exclude that F holds even if neither M not E does (e.g., if the alarm clock is defective).

4. State the probability of the clock failing in terms of the entries of the conditional probability tables of your network (with the deterministic node for the failing clock). 2 pt

Solution: $P(F^+) = 1 - P(M^-) \cdot P(E^-)$

This page was intentionally left blank for extra space

3 Markovian Reasoning

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

- You make daily observations about your business B. Each day business is either good (b_1) or bad (b_2).
- You know this is caused by the general economic situation G, which you cannot easily observe, and which can be getting worse (g_1), be stable (g_2), or getting better (g_3).
- You have previously obtained the following information:
  - when the economy gets worse, your business is good 36% of the time,
  - when the economy is stable, your business is good 84% of the time,
  - when the economy gets better, your business is good 90% of the time,
  - half the time, the economy is the same as on the previous day,
  - when the economy changes from one day to the next, each change is equally likely.

You want to model this situation as a hidden Markov model with two families of random variables indexed by day number d.

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

   Solution: State variables $G_d \in \{g_1, g_2, g_3\}$, evidence variables $B_d \in \{b_1, b_2\}$

2. How can you tell that the sensor model is stationary here? 1 pt

   Solution: The business-economy relation is the same for each day.

3. What order does the model have? 1 pt

   Solution: first-order

4. Complete the following sentence: The transition model T is given by the matrix 2 pt

$$T = \begin{pmatrix} \quad & \quad & \quad \\ & & \\ & & \end{pmatrix} \qquad \text{where} \qquad T_{ij} = P(G_{d+1} = g_j \mid G_d = g_i).$$

   Solution: $T = \begin{pmatrix} 0.5 & 0.25 & 0.25 \\ 0.25 & 0.5 & 0.25 \\ 0.25 & 0.25 & 0.5 \end{pmatrix}$

5. Complete the following sentence: The sensor model S is given by the matrix 2 pt

$$S = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix} \qquad \text{where} \qquad S_{ij} = P(B_d = b_j \mid G_d = g_i).$$

Solution: $S = \begin{pmatrix} 0.36 & 0.64 \\ 0.84 & 0.16 \\ 0.9 & 0.1 \end{pmatrix}$

6. Let T be as above and let $\mathbf{v}$ be a 3-dimensional vector whose coefficients sum to 1. What is the intuitive meaning of the property $T \cdot \mathbf{v} = \mathbf{v}$? 2 pt

Solution: $\mathbf{v}$ is a probability distribution of the economy that is a fixed point of the transition model, i.e., the distribution will stay the same when predicting the future.

7. Assume you want to apply filtering after observing good business at $t = 1$. Give the diagonal sensor matrix O_1 to use in this case. 1 pt

Solution: $O_1 = \begin{pmatrix} 0.36 & 0 & 0 \\ 0 & 0.84 & 0 \\ 0 & 0 & 0.9 \end{pmatrix}$

Problem 3.2 (Decision Processes and Utility)
Consider a robot arranging 9 identical objects into a frame with 9 identical slots. Initially, all objects are stacked next to the frame.
Each object can be placed into every one of the 9 slots in two ways: correctly aligned or mis-aligned. The robot's task is to place one correctly aligned object into each slot.
In each move, the agent can do one of the following:

- Put the next unplaced object into a free slot. There is a 10% chance the placement is mis-aligned.
- Remove a mis-aligned object from the frame and add it back to the stack.
- Do nothing.

1. Model this situation as a Markov Decision Process $\langle S, A, T, s_0, R\rangle$. Use a reward function that uses -0.1 for non-goal states. 5 pt

---

Solution: One possible model is

- $S = \{F, A, M\}^9$ where $F/A/M$ represents the each slot as free/aligned/mis-aligned. Below we write $s^{i=e}$ for the state $(s_1, \ldots, s_{i-1}, e, s_{i+1}, \ldots, s_9)$.
- $A(s) = \{N\} \cup \{P_i \mid s_i = F\} \cup \{R_i \mid s_i = M\}$ with N for nothing, P_i for "place in slot i", and R_i for "remove from slot i"
- The transition model is given by
  - $P(s \mid s, N) = 1$
  - $P(s^{i=A} \mid s, P_i) = 0.9$ and $P(s^{i=M} \mid s, P_i) = 0.1$
  - $P(s^{i=F} \mid s, R_i) = 1$

  and all other probabilities are 0.
- $s_0 = (F, \ldots, F)$
- $R((A, \ldots, A)) = 1$ and $R(s) = -0.1$ otherwise

---

2. Give an optimal policy π^*. 2 pt

Solution: Any policy is optimal that places into free positions, removes mis-aligned pieces, and does nothing when finished. E.g.: $\pi^*(s) = R_i$ for some i with $s_i = M$; or, if no such i exists, $\pi^*(s) = P_i$ for some i with $s_i = F$; and $\pi^*(s) = N$ otherwise.

---

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

---

Solution: $U(s) = R(s) + 0.5 \max_{a\in A(s)} \Sigma_{s'\in S} U(s') \cdot P(s' \mid s, a)$
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: In the initial state, every action leads to a state with utility r where r is the constant negative reward, e.g., $r = -0.1$. So we have $U(s_0) = 1.5 \cdot r$.

---

4. Following up on the previous question: What is the smallest number of iterations, after which the utility of the initial state can be positive? 1 pt

---

Solution: 9 (After 1 iteration, only the goal state is positive. Each iteration makes at most the next state along the path to the goal positive.)

---

4 Learning

Problem 4.1 (Decision Trees)
Consider an unknown natural number $N \in \{0, \ldots, 3\}$ that we want to determine from certain Boolean attributes. For example, knowing that N is positive and even determines that $N = 2$.

1. Give the entropies $I(positive)$ and $I(even)$. Which of the two has higher information gain? 3 pt

   Solution: $I(even) = 1$
   $I(positive) = -1/4 \log 1/4 - 3/4 \log 3/4$
   even has higher gain.

2. Define 3 Boolean-valued attributes of N such that the following hold: 3 pt
   - The smallest decision tree for N using these attributes has depth 2.
   - There is no decision tree for N that uses only 2 of the 3 attributes.

   Solution: For example, *even* (true for 0, 2), *isZero* (true for 0), and *isOne* (true for 1).

3. Now assume our goal is to learn the function that computes whether N is a square of a natural number, using the attributes *positive* and *even*. We use $N = 0$ and $N = 1$ as examples. Formally state this situation as an inductive learning problem $\langle \mathcal{H}, T \rangle$. 2 pt

   Solution: $\mathcal{H}$ is the set of functions $\mathbb{B} \times \mathbb{B} \to \mathbb{B}$. T is the set containing $((no, yes), yes)$ and $((yes, no), yes)$.
   Alternatively, any subset of the hypothesis space can be used.

Problem 4.2 (Statistical Learning)
You observe the values below for 50 games of a tennis player. You want to predict the result based on time of day and opponent.

		Number of	
Time	Opponent	wins	losses
Morning	Weaker	5	1
Afternoon	Weaker	6	2
Evening	Weaker	3	0
Morning	Similar	3	3
Afternoon	Similar	2	3
Evening	Similar	4	5
Morning	Stronger	2	2
Afternoon	Stronger	1	3
Evening	Stronger	1	4

1. What is the hypothesis space for this classification task, seen as a decision tree learning problem? 2 pt

Solution: The set of functions

$$\{Morning, Afternoon, Evening\} \times \{Weaker, Similar, Stronger\} \to \{Win, Loss\}$$

2. Explain (in about 2 sentences) the key characteristic of this data that makes it difficult to use decision tree–based classification methods from the lecture and why. 2 pt

Solution: The results are not uniquely determined by the input. So no decision tree can produce the classification. If each row had a 0 in one of the Win/Loss columns, it would work.

3. Now instead, consider this as a statistical learning problem. As hypotheses, we use the probability distributions $P(Result \mid Time, Opponent)$. 2 pt
Relative to the observed data, give the likelihood of the following hypothesis: The player's probability to win is 80% if the game is against a weaker player, and it is 10% otherwise.

Solution: The likelihood is the probability of the data under the condition that the hypothesis holds. This is $0.8^{14} \cdot 0.2^{3} \cdot 0.1^{13} \cdot 0.9^{20}$

4. To learn a hypothesis via Bayesian learning, we model this situation as a Bayesian network $Time \to Result \leftarrow Opponent$. Give the resulting entries of the conditional probability table for 2 pt
   1. $P(Opponent = Weaker) =$
   2. $P(Result = win \mid Time = Afternoon, Opponent = Weaker) =$

Solution: $P(Opponent = Weaker) = 0.34$
$P(Result = win \mid Time = Afternoon, Opponent = Weaker) = 0.75$

Problem 4.3 (Inductive Learning)
Assume we already know the predicate $\mathtt{par}(x, y)$ for x being a parent of y. Our goal is to learn the predicate $\mathrm{gp}(x, y)$ for x being a grandparent of y, i.e., to find a first-order formula D such that $\forall x, y.\mathrm{gp}(x, y) \Leftrightarrow D(x, y)$.
We do not know D, but we have the following (counter-)examples for gp:

x	y	$\mathrm{gp}(x, y)$
A	H	yes
B	I	yes
A	E	no
A	F	no
A	B	no
A	C	no

1. Give the first-order formula D that represents $\mathrm{gp}(x, y)$. 2 pt

Solution: $D(x, y) = \exists u.\mathtt{par}(x, u) \wedge \mathtt{par}(u, y)$

2. Explain (in about 2 sentences) the key advantages of using inductive learning to learn a formula for gp as opposed to using the formula that can be derived from a decision tree for the example set. 2 pt

Solution: Inductive learning can learn more complex first-order formulas, in particular the ones using additional quantifiers like in the correct formula for D. Inductive learning can take the background knowledge into account such as referring to `par`, which is decision trees cannot do.
The formula obtained from a decision tree tends to encode all examples in a big formula. That usually takes a lot more space. It also overfits extremely: the learned formula would not be able to correctly answer examples that were not part of the training set.

3. Assume that a first-order inductive learning algorithm has partially learned the predicate as the clause 2 pt

$$\mathrm{gp}(x, y) \Leftarrow \mathtt{par}(x, u) \wedge \ldots$$

Give two options of non-recursive literals that the algorithm might reasonably try next to complete the clause.

Solution: Any application of `par` to x, y, u, v except $\mathtt{par}(v, v)$ (all variables new) or $\mathtt{par}(x, u)$ (already used).

5 Natural Language Processing

Problem 5.1 (Language Models)

1. How many different trigrams does a language with n words have? 1 pt

   Solution: n^3

2. What is a statistical language model? 2 pt

   Solution: A probability distribution over words or n-grams occurring in texts for the language.

3. Name two applications of statistical language models. 2 pt

   Solution:

4. Why is it work-intensive in practice to build a good statistical language model for a natural language? 2 pt

   Solution: Because a large and representative corpus of texts has to be aggregated and processed. That takes time/effort.

Problem 5.2 (Seq2Seq Translation)

1. Explain (in about 2 sentences) the key idea of seq2seq models for machine translation. 2 pt

---

Solution: It concatenates two neural networks, one to encode the source, followed by one to decode into the target language. Inputs are fed into the encoder token-wise. When the input it processed, the decoder generates outputs one word at a time.

---

2. Explain (in about 2 sentences) the role of beam search in the decoder. 2 pt

---

Solution: It searches through possible decodings. Rather than committing to an output word in each step, each step keeps the top k hypotheses for the output sequence.

This page was intentionally left blank for extra space