UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
CS440/ECE448 Artificial Intelligence
Exam 5
Spring 2026

May 12, 2026, 8:00

Your Name: ____

Your NetID: ____

Instructions

- SHOW YOUR WORK! Correct numerical answers will not receive full credit unless you show sufficient derivation to convince us that you understand the answer.
- Have your ID ready; you will need to show it when you turn in your exam.
- Make sure that your answer includes only the variables that it should include, but DO NOT simplify explicit numerical expressions, including expressions involving standard functions (exp, log, sin, cos) and expressions involving the sum, product, maximum, or argmax of a complete list of alternatives. For example, the answer $x = \max\left(\sin(0.3), \frac{1}{1+\exp(-0.1)}\right)$ is MUCH preferred (much easier for us to grade) than the answer $x = 0.524979$.
- This will be a CLOSED BOOK, CLOSED NOTES exam. You are permitted to bring and use TWO 8.5x11 pages of notes, front and back, handwritten.
- No electronic devices (phones, tablets, calculators, computers etc.) are allowed.

Possibly Useful Formulas

$$P(X=x|Y=y)P(Y=y) = P(Y=y|X=x)P(X=x)$$
$$P(X=x) = \sum_y P(X=x, Y=y)$$
$$E\left[f(X,Y)\right] = \sum_{x,y} f(x,y)P(X=x, Y=y)$$

Precision,Recall=Sensitivity,Specificity $= \frac{TP}{TP+FP}, \frac{TP}{TP+FN}, \frac{TN}{TN+FP}$

MPE=MAP: $f(x) = \arg\max\left(\log P(Y=y) + \log P(X=x|Y=y)\right)$

Naive Bayes: $P(X=x|Y=y) \approx \prod_{i=1}^{n} P(W=w_i|Y=y)$

Laplace Smoothing w/OOV: $P(W=w_i) = \frac{k + \text{Count}(W=w_i)}{k + \sum_v (k + \text{Count}(W=v))}$

Fairness: $P(Y|A) = \frac{P(Y|f(X),A)P(f(X)|A)}{P(f(X)|Y,A)}$

Score: $v_{t+1}(j) = \max_i v_t(i) + \log a_{ij} + \log b_j(x_{t+1})$

Backpointer: $\psi_{t+1}(j) = \arg\max_i v_t(i) + \log a_{ij} + \log b_j(x_{t+1})$

Linear Regression: $\mathscr{L} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}_i - y_i\right)^2$

$$\sigma(z) = \frac{1}{1+e^{-z}}, \quad \sigma'(z) = \sigma(z)\left(1-\sigma(z)\right)$$
$$f = g\left(\mathbf{w}^T\mathbf{x}\right), \quad \frac{\partial f}{\partial \mathbf{w}} = g'\left(\mathbf{w}^T\mathbf{x}\right)\mathbf{x}$$

$$\text{softmax}(\mathbf{z}) = \frac{\exp(\mathbf{z})}{\sum_i \exp(z_i)}, \quad \frac{\partial\, \text{softmax}_k(\mathbf{z})}{\partial \mathbf{z}} = \underset{k}{\text{softmax}}(\mathbf{z})\left(\mathbb{1}_k - \text{softmax}(\mathbf{z})\right)$$
$$f[k] = \sum_i w[i]x[k-i], \quad \frac{\partial f[k]}{\partial w[i]} = x[k-i]$$

$$\mathbf{C} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}$$
$$S: \{\mathscr{V}_P, \mathscr{V}_Q\} \rightarrow \{\mathscr{V}_Q, \mathscr{C}\} \text{ s.t. } S(P) = S(Q) = U$$
$$\frac{x'}{x} = \frac{y'}{y} = \frac{-f}{z}$$

$$\textbf{Layer-Wise Relevance Propagation: } R(x_d) = \frac{\frac{\partial \mathscr{L}}{\partial x_d} x_d}{\sum_{d'} \frac{\partial \mathscr{L}}{\partial x_{d'}} x_{d'}} R(\mathscr{L})$$

$$\textbf{Configuration Space Obstacles: } \mathscr{C}_{\text{obs}} = \{\mathbf{c} : \exists \mathbf{b} : \phi(\mathbf{b}, \mathbf{c}) \in \mathscr{W}_{\text{obs}}\}$$

$$\textbf{Admissible: } \hat{h}(n) \leq h(n)$$

$$\textbf{Consistent: } \hat{h}(n) - \hat{h}(m) \leq d(n,m)$$

$$\textbf{Bellman: } u(s) = r(s) + \gamma \max_a \sum_{s'} P(s'|s,a) u(s')$$

$$\textbf{Policy Eval: } u_i(s) = r(s) + \gamma \sum_{s'} P(s'|s,\pi_i(s)) u_i(s')$$

$$\textbf{Policy Update: } \pi_{i+1}(s) = \arg\max_a \sum_{s'} P(s'|s,a) u_i(s')$$

$$\textbf{Value Iteration: } u_{i+1}(s) = r(s) + \gamma \max_a \sum_{s'} P(s'|s,a) u_i(s')$$

$$\textbf{Model-based: } P(s_{t+1}|s_t,a_t) = \frac{N(s_t,a_t,s_{t+1}) + k}{\sum_{s'} (N(s_t,a_t,s') + k)}$$

$$\textbf{Q-Learning: } q(s_t,a_t) \leftarrow (1-\eta) q(s_t,a_t) + \eta q_{\text{local}}(t)$$

$$\textbf{TD-Learning: } q_{\text{local}}(t) = r_t + \gamma \max_a q_t(s_{t+1},a)$$

$$\textbf{SARSA: } q_{\text{local}}(t) = r_t + \gamma q_t(s_{t+1},a_{t+1})$$

$$\textbf{Deep Q: } \mathscr{L}_{\text{critic}} = \frac{1}{2} \left(q_t(s_t,a_t) - q_{local}(t)\right)^2$$

$$\textbf{Actor-Critic: } \mathscr{L}_{\text{actor}} = -\sum_a \pi_a(s_t) q_t(s_t,a)$$

$$\textbf{REINFORCE: } \Delta \mathbf{w} = \eta (r - \mu) \sum_t \frac{\partial \log \pi_{a_t}(s_t)}{\partial \mathbf{w}}$$

$$\textbf{Max: } v = \max\left(v, \max_c \left(\mathbb{E}\left[v(c)\right]\right)\right), \quad \alpha = \max\left(\alpha, \max_c \left(\mathbb{E}\left[v(c)\right]\right)\right), \beta = \text{Unchanged}$$

$$\textbf{Min: } v = \min\left(v, \min_c \left(\mathbb{E}\left[v(c)\right]\right)\right), \quad \alpha = \text{Unchanged}, \beta = \min\left(\beta, \min_c \left(\mathbb{E}\left[v(c)\right]\right)\right)$$

$$\textbf{Chance: } \mathbb{E}[v] = \sum_c P(c) v(c), \quad \alpha = \text{Unchanged}, \beta = \text{Unchanged}$$

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix}^T \mathbf{A}\beta = 0, \quad \alpha^T \mathbf{B} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = 0$$

$$a = a + \eta \frac{\partial E[r_A]}{\partial a}, \quad b = b + \eta \frac{\partial E[r_B]}{\partial b}$$

Question 1 ***(24 points)***

A factory produces two types of chips: Type A (70%) and Type B (30%). Type A chips have a 5% defect rate. Type B chips have a 15% defect rate.

(a) (12 points) A randomly selected chip is found to be defective. What is the probability that the defective chip came from Type B?

Solution:

$$P(B|D) = \frac{(0.3)(0.15)}{(0.3)(0.15)+(0.7)(0.05)}$$

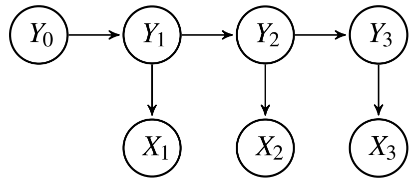
(b) (12 points) Suppose you want to create a naive Bayes classifier to determine whether a chip is of type A or type B. Suppose that you have a training corpus containing N_A chips of type A, of which M_A are cobalt, and N_B chips of type B, of which M_B are cobalt. If a cobalt chip has a defect, under what conditions would you classify that chip as type A? Your answer should be a function of N_A, M_A, N_B, M_B, and the Laplace smoothing parameter k.

Solution: Classify the chip as type A if

$$(0.7)(0.05)\left(\frac{M_A+k}{N_A+2k}\right) > (0.3)(0.15)\left(\frac{M_B+k}{N_B+2k}\right)$$

Question 2 ***(24 points)***

Consider the following hidden Markov model:



Suppose that Y_t and X_t are both binary for all t, with transition and observation probabilities abbreviated as $P(Y_t = j|Y_{t-1} = i) = a_{i,j}$ and $P(X_t = j|Y_t = i) = b_{i,j}$, and suppose it is known that $Y_0 = 0$.

(a) (12 points) In terms of the model parameters, what is $P(Y_1 = 1|Y_0 = 0, Y_2 = 0)$?

Solution:

$$P(Y_1 = 1|Y_0 = 0, Y_2 = 0) = \frac{a_{0,1}a_{1,0}}{a_{0,1}a_{1,0} + a_{0,0}^2}$$

(b) (12 points) Let $v_t(j)$ and $\psi_t(j)$ be the Viterbi score and backpointer, respectively (the log probability of the best path ending in state j at time t, and the corresponding previous state). Suppose that $v_2(0) = -25$, $v_2(1) = -7$, and $X_3 = 1$. Under what condition is $\psi_3(1) = 1$? Express your answer in terms of the model parameters.

Solution: $\psi_3(1) = 1$ if $-7 + \log a_{1,1} > -25 + \log a_{0,1}$.

Question 3 ***(24 points)***

The softmax function converts real-valued inputs, $-\infty < z_n < \infty$, into non-negative outputs that add up to one. Besides the standard softmax, there are many other functions that accomplish these goals, e.g.,

$$f_n(\mathbf{z}) = \frac{|z_n|}{\sum_m |z_m|} \tag{1}$$

where z_n and z_m are the n^{th} and m^{th} scalar elements of the vector $\mathbf{z}$.

(a) (12 points) For the function $f_n(\mathbf{z})$ in Eq. (1), find $\partial f_n/\partial z_s$ for both $s = n$ and $s \neq n$.

Solution:

$$\frac{\partial f_n}{\partial z_s} = \begin{cases} -\frac{|z_n|\text{sign}(z_s)}{(\sum_m |z_m|)^2} & s \neq n \\ \frac{\text{sign}(z_n)}{\sum_m |z_m|} - \frac{z_n}{(\sum_m |z_m|)^2} & s = n \end{cases}$$

(b) (12 points) Equation (1) could be used to construct a self-attention layer in which a context vector $\mathbf{c}_i$ is constructed from query, key, and value vectors $\mathbf{q}_i$, $\mathbf{k}_t$, and $\mathbf{v}_t$, all of which have dimension d:

$$\mathbf{c}_i = \sum_t f_t(\mathbf{z})\, \mathbf{v}_t, \tag{2}$$

where $\mathbf{z} = \frac{1}{\sqrt{d}} [\mathbf{k}_1, \mathbf{k}_2, \ldots]^T \mathbf{q}_i$. Express $\partial c_{i,p}/\partial k_{j,r}$ in terms of $\partial f_n/\partial z_s$ for appropriate values of s and n, where $c_{i,p}$ and $k_{j,r}$ are the p^{th} element of $\mathbf{c}_i$ and the r^{th} element of $\mathbf{k}_j$, respectively.

Solution:

$$\frac{\partial c_{i,p}}{\partial k_{j,r}} = \sum_t \frac{1}{\sqrt{d}} \sum_t \frac{\partial f_t}{\partial z_j} q_{i,r} v_{t,p}$$

Question 4 ***(24 points)***

Suppose you are training a logistic regression network, $\sigma(z) = (1+e^{-z})^{-1}$, with the following unusual loss function with respect to a desired output y:

$$\mathscr{L} = (\sigma(z) - y)^2 \tag{3}$$

(a) (12 points) What is $\partial\mathscr{L}/\partial z$? You may not leave $\sigma'(z)$, the derivative of $\sigma(z)$, in your answer; you must replace it with a simpler form, possibly including $\sigma(z)$.

Solution:

$$\frac{\partial\mathscr{L}}{\partial z} = 2(\sigma(z) - y)\sigma(z)(1 - \sigma(z))$$

(b) (12 points) Suppose that z is computed by max-pooling a convolutional network with residual connections,

$$z = \max_i \max_j \left(b[i,j] + \sum_m \sum_n w[i-m, j-n]x[m,n] \right) \tag{4}$$

Use layer-wise relevance propagation to express, in terms of $\partial\mathscr{L}/\partial z$, the relevance of each input pixel $x[m,n]$ to the output $\sigma(z)$. Your answer may include $\partial\mathscr{L}/\partial z$, z, and/or i^* and j^* defined by

$$(i^*, j^*) = \arg\max_i \arg\max_j \sum_m \sum_n w[i-m, j-n]x[m,n] \tag{5}$$

Solution:

$$R(x[m,n]) = \frac{w[i^* - m, j^* - n]x[m,n]}{\sum_{m',n'} w[i^* - m', j^* - n']x[m',n']}$$

Question 5 ***(24 points)***

Suppose the configuration space and workspace of a robot are related by a forward kinematic function: $\mathbf{w} = \phi(\mathbf{c})$, where both $\mathbf{c}$ and $\mathbf{w}$ are real vectors. A path from start to goal is a sequence of states $\{\mathbf{c}_0, \ldots, \mathbf{c}_T\}$ such that the robot takes exactly one second to travel from $\mathbf{c}_t$ to $\mathbf{c}_{t+1}$ along a straight line in configuration space, $\mathbf{w}_0 = \phi(\mathbf{c}_0)$ is the starting point in the workspace, $\mathbf{g} = \phi(\mathbf{c}_T)$ is the goal in the workspace, and T is the (unknown) time required to reach the goal.

(a) (12 points) Suppose you want to find a path that minimizes the integrated total magnitude velocity of the robot's configuration vector,

$$\mathscr{L} = \int_0^T \left\| \frac{\partial \mathbf{c}}{\partial t} \right\| dt, \tag{6}$$

subject to the constraint that the robot does not hit any obstacles, where $\|\cdot\|$ denotes the speed (magnitude of the velocity) in configuration space. How should you define the distance $d(\mathbf{c}_t, \mathbf{c}_{t+1})$ so that Dijsktra's algorithm finds the desired path? Use $d(\mathbf{c}_t, \mathbf{c}_{t+1}) = \infty$ to denote the case when there should be no direct connection from $\mathbf{c}_t$ to $\mathbf{c}_{t+1}$.

> **Solution:**
>
> $$d(\mathbf{c}_t, \mathbf{c}_{t+1}) = \begin{cases} \|\mathbf{c}_{t+1} - \mathbf{c}_t\| & \text{no obstacles between } \mathbf{c}_t, \mathbf{c}_{t+1} \\ \infty & \text{otherwise} \end{cases}$$

(b) (12 points) Suppose that $\hat{h}(\mathbf{c}) = \|\mathbf{c} - \phi^{-1}(\mathbf{g})\|$, Euclidean distance (not squared Euclidean distance!) in configuration space between $\mathbf{c}$ and the goal. Suppose that you put $\mathbf{c}_0$ on the frontier with $g(\mathbf{c}_0) = 0$, then iterate the following loop: (1) From the frontier, pop the node with the lowest $g(\mathbf{c}) + \hat{h}(\mathbf{c})$, (2) Add $\mathbf{c}$ to the explored set, (3) If $\phi(\mathbf{c}) = \mathbf{g}$, end the search and report the path. (4) For each node $\mathbf{s}$ that has $d(\mathbf{c}, \mathbf{s}) < \infty$ and for which $\mathbf{s}$ is not yet in the explored set, add $\mathbf{s}$ to the frontier with $g(\mathbf{s}) = g(\mathbf{c}) + d(\mathbf{c}, \mathbf{s})$. Does this algorithm always find the shortest path? Why or why not? If you are claiming admissibility or consistency, please prove it.

> **Solution:** Yes. This is A* with an explored set, so it's optimal if $\hat{h}(\mathbf{c})$ is consistent, i.e., if $\hat{h}(\mathbf{n}) - \hat{h}(\mathbf{m}) \le d(\mathbf{n}, \mathbf{m})$, but the triangle inequality tells us that this is so:
>
> $$\|\mathbf{n} - \phi^{-1}(\mathbf{g})\| + \|\mathbf{m} - \phi^{-1}(\mathbf{g})\| \le \|\mathbf{n} - \mathbf{m}\|$$

Question 6 ***(24 points)***

Consider an MDP in which the state variable is a real number, $s \in \Re$, and the reward in any state is equal to the value of the state, $r(s) = s$. Suppose there are two possible actions, $a \in \{-1, +1\}$, and suppose the state transition probabilities are

$$P(s'|s,a) = \begin{cases} p & s' = s + 2a \\ q & s' = s - 3a \\ 1 - p - q & s' = s \end{cases}$$

where the parameters p and q are assumed known in part (a), but assumed unknown in part (b).

(a) (12 points) Suppose you try policy iteration with an initial policy that always tries action $\pi_1(s) = +1$, and suppose that policy evaluation results in the estimate $u_1(s) = 50s$. Perform one step of policy update: As a function of the parameters p, q, and/or γ, what is the new policy, $\pi_2(s)$, as a function of s?

Solution:

$$\pi_2(s) = \arg\max_a (50s + 50a(2p - 3q))$$
$$= \begin{cases} +1 & 2p > 3q \\ -1 & \text{otherwise} \end{cases}$$

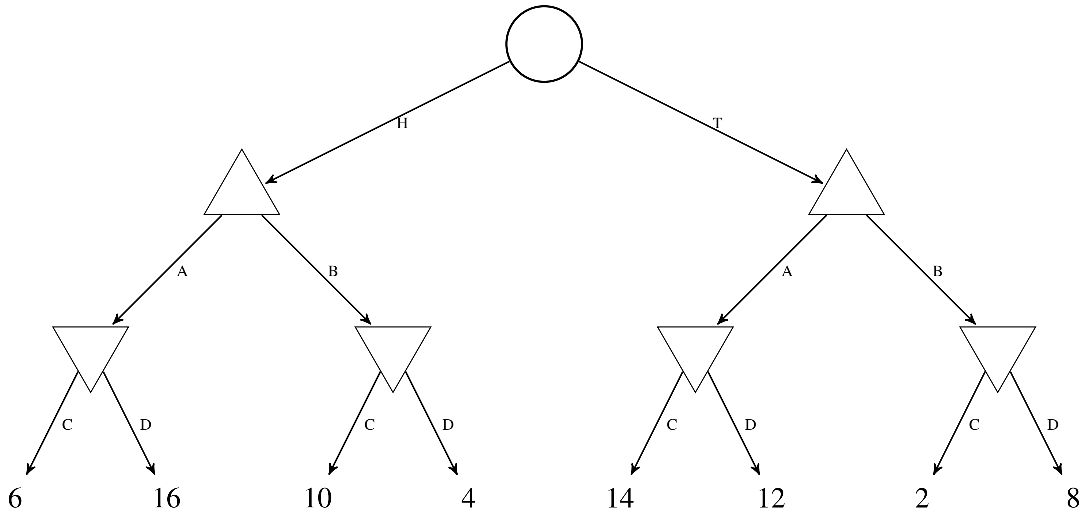
(b) (12 points) Suppose that you are using deep-Q-learning with a simple one-layer neural network, $q(s,a) = \mathbf{w}_T \begin{bmatrix} 1 \\ s \\ a \end{bmatrix}$, where $\mathbf{w} = [w_1, w_2, w_3]^T$ is a weight vector. Starting in state $s_1 = 1$ with $\mathbf{w} = [0, 50, -1]^T$, you perform action $a_1 = 1$, receive a reward of $r_1 = 1$, transition to state $s_2 = -2$, and then perform a single iteration of Q-learning. As a function of the learning rate η and the discount factor γ, what is the new value of $\mathbf{w}$? Estimate q_{local} using SARSA, assuming that your next action will be $a_2 = -1$.

Solution:

$$\mathbf{w} = \begin{bmatrix} 0 \\ 50 \\ -1 \end{bmatrix} - \eta \left([0, 50, -1] \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - 1 + 99\gamma \right) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

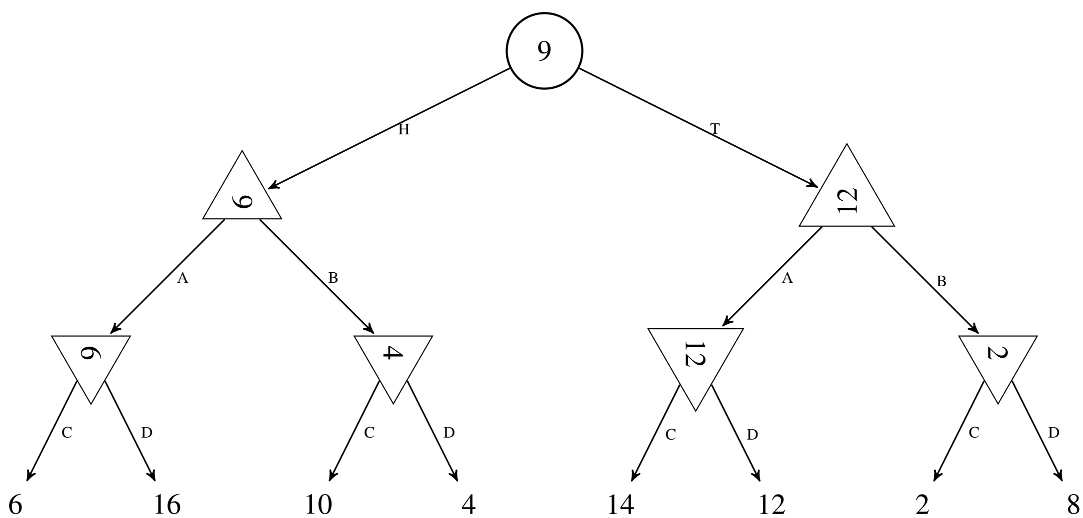
Question 7 ***(24 points)***

Consider the following game tree. First, a fair coin is flipped, and comes up either H or T. Second, Max performs action A or B. Third, Min performs action C or D. Finally, Max receives the number of points shown in the leaf node, and Min receives 9 minus that number of points.



(a) (12 points) What is the value, to Max, of each of the nodes in the above tree? Write the value of each node inside the corresponding node. What actions should Max and Min perform in each node where they are allowed to act?

Solution: Max performs A in both cases; Min's actions are C,D,D,C. The tree is



(b) (12 points) If the above tree is searched using the alpha-beta algorithm, and if the children of each node are searched left-to-right, what values of α and β are inherited by each of the Min nodes? Write, in the table below, the values of α and β that are inherited by each Min node when the algorithm first enters that node. If there is any child of a Min node that alpha-beta will not expand, put an "X" in the corresponding cell of the table below.

α								
β								
Child Node	6	16	10	4	14	12	2	8

Solution:

α	$-\infty$		6		$-\infty$		12	
β	∞		∞		∞		∞	
Child Node	6	16	10	4	14	12	2	X

Question 8 ***(32 points)***

Alice and Bob are playing a simultaneous game in which each of them may either cooperate or defect. Their payoffs are shown in the matrix below.

		Bob	
		defects	cooperates
Alice	defects	-7 11	-3 -12
	cooperates	3 3	-7 16

(a) (11 points) This game has a mixed equilibrium. What is it?

Solution:

$$P(\text{Alice cooperates}) = \frac{2}{7}$$
$$P(\text{Bob cooperates}) = \frac{2}{9}$$

(b) (10 points) How many Pareto-optimal outcomes does this game have, and what are they?

Solution: Two: Both cooperate, or Alice cooperates while Bob defects.

(c) (11 points) It is possible to modify just one of the payoffs in this game (just one number in the chart above) so that Bob has a dominant strategy. What number can be modified, to what value, to give Bob a dominant strategy?

Solution: Any modification that satisfies any of the following four inequalities is a valid solution:

$$r_B(d,d) > -3$$
$$r_B(d,c) < -7$$
$$r_B(c,d) < -7$$
$$r_B(c,c) > 3$$

This page is scratch paper

This page is scratch paper, you may tear it off if you wish