University of Illinois at Urbana-Champaign

CS440/ECE448 Artificial Intelligence

Conflict Exam 5

Spring 2026

Main Exam is May 12, 2026

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: $P(W=w_i) = \frac{k+\text{Count}(W=w_i)}{k+\sum_v\left(k+\text{Count}(W=v)\right)}$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Suppose you are a doctor, trying to determine if your patient has a particular disease. The disease has a 2% prevalence (2% of the population has the disease).

(a) (12 points) Your patient takes a blood test to determine if they have the disease. The test sensitivity (true positive rate) is 95%, and its specificity (true negative rate) is 90%. Your patient tests positive. What is the posterior probability that the patient actually has the disease?

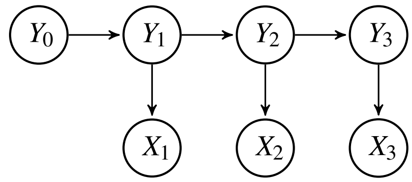
> **Solution:**
>
> $$P(Y=1|\hat{Y}=1)=\frac{(0.02)(0.95)}{(0.02)(0.95)+(0.98)(0.1)}$$

(b) (12 points) Now suppose that the test does not exist, and you must invent it. Of people who have the disease, 80% have symptom A, and 70% have symptom B. In a training database of N_0 people without the disease, N_A have symptom A, and N_B have symptom B. If a person has both symptom A and symptom B, under what circumstances would naive Bayes specify that they have the disease? Your answer should be a function of N_0, N_A, N_B, and the Laplace smoothing parameter k.

> **Solution:** Detect the disease if
>
> $$(0.02)(0.8)(0.7)>(0.98)\left(\frac{N_A+k}{N_0+2k}\right)\left(\frac{N_B+k}{N_0+2k}\right)$$

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 $P(Y_1 = 0|Y_0 = 0, X_1 = 1)$?

> **Solution:**
>
> $$P(Y_1 = 0|Y_0 = 0, X_1 = 1) = \frac{a_{0,0}b_{0,1}}{a_{0,0}b_{0,1} + a_{0,1}b_{1,1}}$$

(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 $v_2(0) = -8$, $v_2(1) = -15$, and $X_3 = 0$. Under what condition is $\psi_3(0) = 0$? Express your answer in terms of the model parameters.

> **Solution:** $\psi_3(0) = 0$ if $-8 + \log a_{0,0} > -15 + \log a_{1,0}$.

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.,

$$f_n(\mathbf{z}) = \frac{z_n^4}{\sum_m z_m^4} \tag{1}$$

where z_n and z_s are the n^{th} and s^{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:

$$\frac{\partial f_n}{\partial z_s} = \begin{cases} -\frac{4 z_n^4 z_s^3}{\left(\sum_m z_m^4\right)^2} & s \neq n \\ \frac{4 z_n^3}{\sum_m z_m^4} - \frac{4 z_n^7}{\left(\sum_m z_m^4\right)^2} & s = n \end{cases}$$

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

Solution:

$$\frac{\partial c_{i,p}}{\partial v_{j,r}} = \begin{cases} 0 & p \neq r \\ f_j(\mathbf{z}) & p = r \end{cases}$$

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:

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

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

(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 offset input term $b[i,j]$ to the output $\sigma(z)$. Your answer may include $\partial\mathscr{L}/\partial z$, z, and/or i^* and j^* defined by

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

> **Solution:**
>
> $$R(b[i,j]) = \begin{cases} 1 & i = i^*, j = j^* \\ 0 & \text{otherwise} \end{cases}$$

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 $\mathbf{c}$ and $\mathbf{w}$ are both 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 total time taken for the robot to travel from start to goal,

$$\mathscr{L} = T \tag{6}$$

subject to the constraint that the robot does not hit any obstacles. 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:

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

(b) (12 points) Suppose that $\hat{h}(\mathbf{c}) = 1$ for all non-goal nodes, and $\hat{h}(\mathbf{g}) = 0$. 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 this is trivially true since $\hat{h}(\mathbf{n}) - \hat{h}(\mathbf{m}) = 0$ in every case except when $\mathbf{m}$ is the goal, and even in that case, $\hat{h}(\mathbf{n}) - \hat{h}(\mathbf{m}) = 1 \le d(\mathbf{n}, \mathbf{m})$.

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 $\pi_1(s) = +1$. Policy evaluation in this case results in equations of the form $s = \sum_k a_k u_1(s+k)$, where $u_1(s)$ is estimated utility of state s. Find the coefficients of these equations, a_k, in terms of p, q, and/or γ.

Solution: Simplifying the policy evaluation equation, we get

$$s = u_1(s)(1 - \gamma(1 - p - q)) - u_1(s+2)\gamma p - u_1(s-3)\gamma q,$$

so $a_0 = 1 - \gamma(1 - p - q)$, $a_{-3} = -\gamma q$, $a_2 = -\gamma p$, and $a_k = 0$ for all other k.

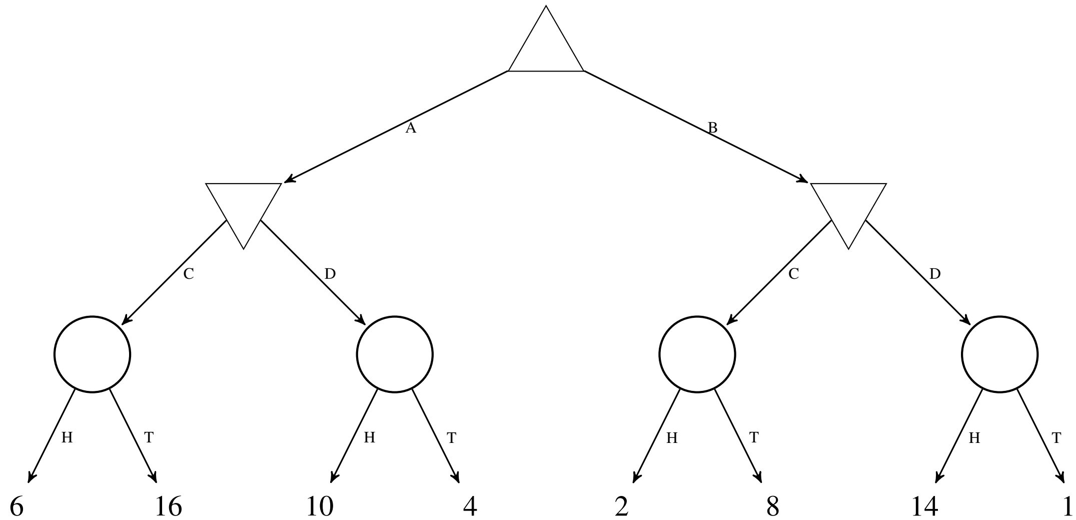
(b) (12 points) Suppose that you are using deep-Q-learning with a simple one-layer neural network, $q(s,a) = \mathbf{w}^T \begin{bmatrix} 1 \\ s \\ a \end{bmatrix}$, where $\mathbf{w} = [w_1, w_2, w_3]^T$ is a weight vector. Starting in state $s_1 = 1$ with $\mathbf{w} = [0, 1, 1]^T$, you perform action $a_1 = 1$, 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 TD-learning.

Solution:

$$\mathbf{w} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - \eta(1+\gamma) \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

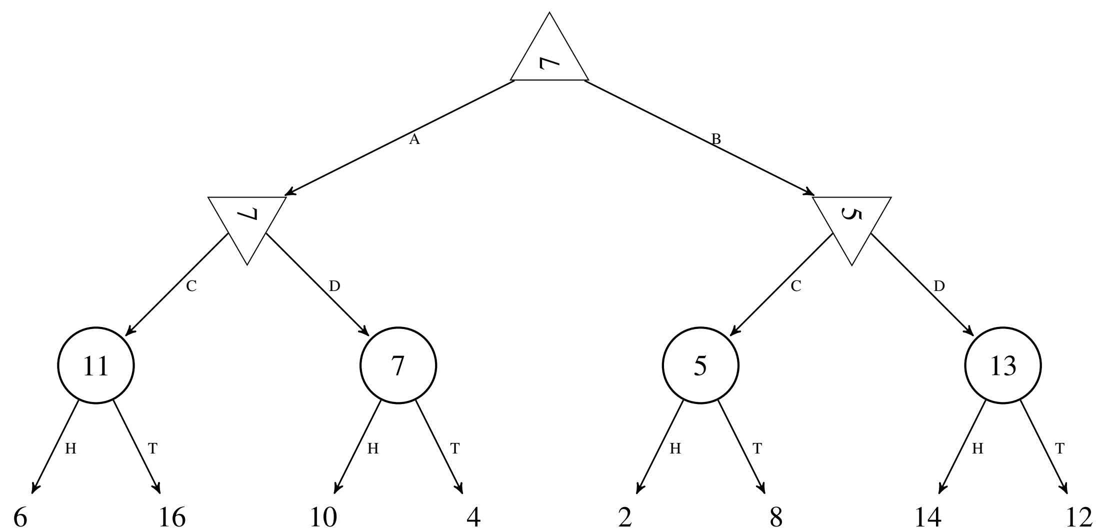
Question 7 ***(24 points)***

Consider the following game tree. First, Max performs either action A or B. Second, Min performs action C or D. Third, A fair coin is flipped, and comes up either H or T. 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 plays A. Regardless of what Max plays, Min plays D. 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 that node when the algorithm first enters that node. If there is any child of a Min node that alpha-beta will not expand, enter an an "X" in that cell of the table below.

α				
β				
Child Node	Chance(6,16)	Chance(10,4)	Chance(2,8)	Chance(14,12)

Solution:

α	$-\infty$		7	
β	∞		∞	
Child Node	Chance(6,16)	Chance(10,4)	Chance(2,8)	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:

$$P(\text{Alice cooperates}) = \frac{5}{7}$$
$$P(\text{Bob cooperates}) = \frac{7}{9}$$

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

Solution: Two: Alice cooperates and Bob defects, or vice versa.

(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