CS 188 Spring 2025

Intro to Artificial Intelligence Final Exam

PRINT Your Name: ______________________________

PRINT Your Student ID: ______________________________

PRINT Student name to your left: ______________________________

PRINT Student name to your right: ______________________________

You have 170 minutes. There are 9 questions of varying credit. (100 points total)

Question:	1	2	3	4	5	6	7	8	9	Total
Points:	10	9	9	14	15	11	8	14	10	100

We reserve the right to deduct points for failing to follow the marking directions below:

For questions with **circular bubbles**, you may select only one choice.

- ○ Unselected option (Completely unfilled)
- ☑ Don't do this (it will be graded as incorrect!)
- ● Only one selected option (completely filled)

For questions with **square checkboxes**, you may select one or more choices.

- ■ You can select
- ■ multiple squares
- ☑ Don't do this (it will be graded as incorrect!)

Anything you write outside the answer boxes or you ~~cross out~~ will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

Read the honor code below and sign your name.

By signing below, I affirm that all work on this exam is my own work. I have not referenced any disallowed materials, nor collaborated with anyone else on this exam. I understand that if I cheat on the exam, I may face the penalty of an "F" grade and a referral to the Center for Student Conduct.

SIGN your name: ______________________________

Q1 *Counting Calories* **(10 points)**

Consider a variant of Pacman where each food dot has a different *calorie count*. Each food dot's calorie count is a strictly positive integer. Each food dot that Pacman eats adds to his total calorie count (C).

When Pacman moves into a square with a dot, he automatically eats the dot as part of that action. All actions cost 1.

Pacman's goal is to eat at least 10 calories in total, i.e. reach a state where $C \geq 10$.

Q1.1 (3 points) Select all admissible heuristics for this problem.

☐ C

☐ $10 - C$

☐ Manhattan distance to the nearest dot.

☐ Sum of Manhattan distances to all dots.

☐ Euclidean distance to the nearest dot.

☐ Sum of Euclidean distances to all dots.

○ None of the above

For each of the next three subparts, consider the given maze and search algorithms. Select whether each search algorithm will find the optimal solution with **at most 10 states** expanded.

Hint: Recall that expanding a state means calling the successor function on that state.

For all subparts, when calling the successor function, enqueue the Left successor state first, then the Right successor state. This means that for DFS, the Right successor state gets popped off the stack first.

Q1.2 (2 points) A 1×11 grid with 10 dots, each worth 1 calorie. Pacman starts in the leftmost square.

P	•	•	•	•	•	•	•	•	•	•

☐ DFS tree search ☐ BFS tree search ○ None of the above

Q1.3 (2 points) A 1×11 grid with 10 dots, each worth 1 calorie. Pacman starts in the middle square.

•	•	•	•	•	P	•	•	•	•	•

☐ DFS tree search ☐ BFS tree search ○ None of the above

Q1.4 (2 points) A 1×5 grid with a single 10-calorie dot. The dot is at the right, and Pacman is at the left.

P				●

☐ DFS tree search ☐ BFS tree search ○ None of the above

Q1.5 (1 point) UCS where all actions cost 1 is always equivalent to which search algorithm?

○ BFS ○ DFS ○ A* ○ None

Q2 *Constrained Staff Photos* **(9 points)**

Four CS 188 TAs want to line up to take staff photos, and Pacman wants to assign each TA to one position.

There are 6 possible positions for TAs to stand, numbered 1 through 6 from left to right. The TAs are the variables, and the positions are the values.

No two TAs can stand in the same position. Not all positions need to be assigned to a TA.

The TAs have special requests:

- Advika (A) does not want anyone to stand directly to her left or right.
- Josh (J) does not want to stand in the leftmost or rightmost position.
- Michael (M) wants to stand directly next to Josh.
- Saathvik (S) does not want to stand directly next to Josh.

Two example assignments are shown below. The invalid assignment violates Advika's, Josh's, and Michael's requests, though Saathvik's request is satisfied.

Valid:

A		M	J		S
1	2	3	4	5	6

Invalid:

J			S	M	A
1	2	3	4	5	6

Q2.1 (1 point) Which type of constraint can be used to represent Advika's request in a single constraint?

○ Unary ○ Binary ○ Higher-order

Q2.2 (1 point) Which type of constraint can be used to represent Josh's request in a single constraint?

○ Unary ○ Binary ○ Higher-order

Q2.3 (2 points) For this subpart, no variables are assigned, and every variable's domain is $\{1, 2, 3, 4, 5, 6\}$.

Pacman uses LCV (least constraining value) with forward checking to assign Advika first. Select all value(s) that LCV could assign to A.

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

For the next two subparts, consider the partial assignment and domains below:

					A
1	2	3	4	5	6

$J : \{2, 3, 4\}$ $M : \{1, 3, 4\}$ $S : \{1, 3, 4\}$

Q2.4 (2 points) Pacman runs arc consistency to check the arc $(S \to J)$.

Select all values that **remain** in the domain of S.

☐ 1 ☐ 3 ☐ 4 ○ None of the above

Q2.5 (3 points) Suppose Pacman runs the arc consistency algorithm covered in lecture (AC-3). Select all arcs that are enqueued after processing the $(S \to J)$ arc.

☐ $(J \to S)$ ☐ $(J \to M)$ ☐ $(M \to S)$

☐ $(S \to J)$ ☐ $(M \to J)$ ☐ $(S \to M)$ ○ None of the above

Q3 *Game o' f(Thrones)* **(9 points)**

Pranav and Catherine are playing a game. Pranav acts as a standard maximizer (represented by triangles), and Catherine is represented by hexagonal nodes. Catherine chooses an action that **minimizes** some function f applied to each of the child nodes.

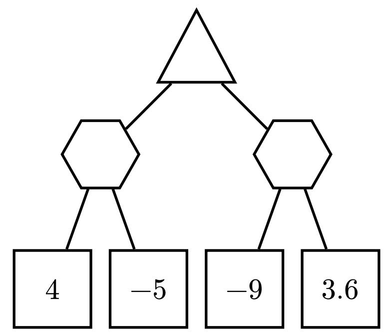
For example:

- If $f(x) = x$, then Catherine acts as a standard minimizer.
- If $f(x) = -x$, then Catherine acts as a standard maximizer.
- If $f(x) = 0$, then Catherine randomly selects an action, since she will see each of her children as having an equal value of 0.

Assumptions:

- Pranav always knows Catherine's strategy and acts optimally.
- Terminal nodes are labeled with Pranav's utility, not Catherine's utility.
- Unless otherwise stated, terminal nodes are unbounded (can take any real value).

Q3.1 (1 point) For this subpart only, consider the tree to the right:



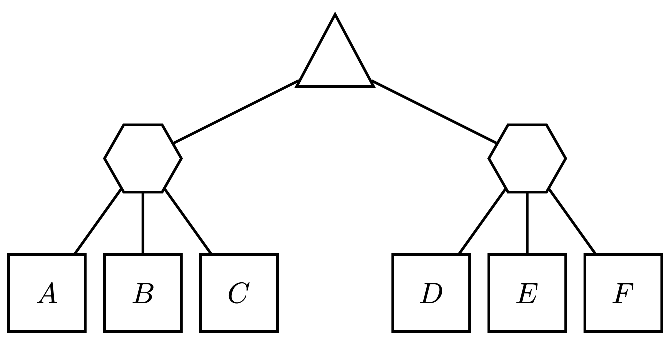
If Catherine chooses an action that minimizes the function

$$f(x) = \begin{cases} 1 \text{ if } x \geq 0 \\ 0 \text{ if } x < 0 \end{cases}$$

what utility does Pranav receive in this game tree, assuming he acts optimally?

○ 4 ○ −5 ○ −9 ○ 3.6

Q3.2 (2 points) For this subpart only, consider the tree below:



If Catherine chooses an action that minimizes the function

$$f(x) = (x - 188)^2$$

what values of A make it possible to prune B? If no values of A cause B to be pruned, leave both boxes blank and bubble None.

Assume that nodes are evaluated from left to right, and we prune on equality.

[ ] $\leq A \leq$ [ ] ○ None

For the remaining subparts, consider any game tree with alternating maximizer and hexagonal nodes, not necessarily the tree above.

Assumptions for the remaining subparts:
- Both Pranav and Catherine evaluate the entire game tree (no depth-limiting).
- No alpha-beta pruning takes place.

Q3.3 (2 points) Catherine minimizes the same function f as above:

$$f(x) = (x - 188)^2$$

For some given game tree, Pranav's utility when Catherine minimizes $f(x) = (x - 188)^2$ is ________ strictly less than Pranav's utility when Catherine is a standard minimizer.

○ always ○ sometimes ○ never

Q3.4 (1 point) Which type of game can model this scenario for **all** functions f?

○ Minimax ○ Multi-Agent Utilities

○ Expectimax ○ None of the above

Q3.5 (3 points) For this subpart only, assume the values at all leaf nodes are strictly positive.

Which of the following functions will cause Catherine to always act like a standard minimizer? Select all that apply.

☐ $\frac{1}{x}$ ☐ $\mathrm{ReLU}(x)$ ☐ $x + 5$

☐ $\log(x)$ ☐ e^x ☐ $x - 5$

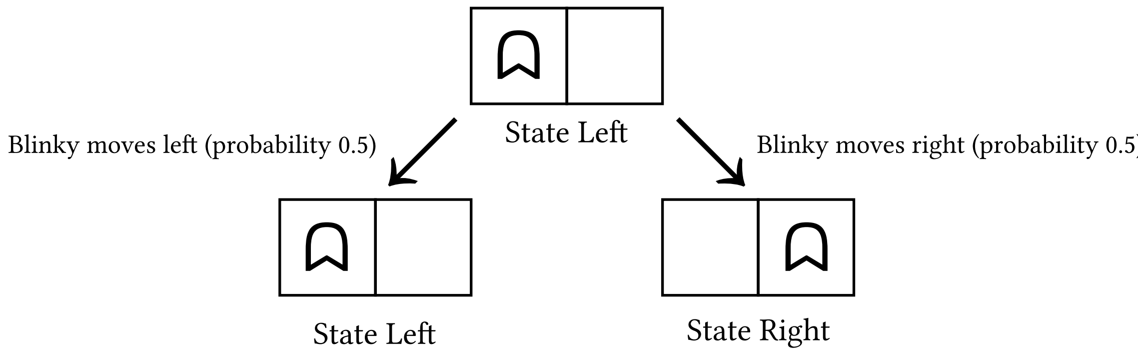
○ None of the above

Q4 *Busted!* **(14 points)**

Consider a 1×2 grid where Blinky can be in the left or right square. At each time step, Blinky will move left with probability 0.5 or move right with probability 0.5. If the move would result in Blinky moving off the grid, Blinky stays in the same position.

State **`Left`** represents Blinky in the left square, and state **`Right`** represents Blinky in the right square.

An example is shown below that starts in state **`Left`**.



Pacman is trying to defeat Blinky and has 3 possible actions: **`Bust Left`**, **`Bust Right`**, and **`Don't Bust`**. Note that Pacman does not have control over Blinky.

`Bust Left` and **`Bust Right`** give a reward of $+1$ for busting correctly (the same square as Blinky), and -1 for busting incorrectly. The **`Bust Left`** and **`Bust Right`** actions transition into a terminal state **`X`** where no further actions or rewards are available, regardless of whether the bust was correct.

The **`Don't Bust`** action gives a reward of 0. Then, Blinky moves left or right, and Pacman can take another action.

Q4.1 (4 points) Fill in the table for the transition and reward functions (some rows have been omitted). If a transition (s, a, s') occurs with probability 0, write "N/A" in the box for $R(s, a, s')$.

s	a	s'	$T(s, a, s')$	$R(s, a, s')$
`Left`	`Bust Left`	`Left`		
`Left`	`Bust Left`	`Right`		
`Left`	`Bust Left`	`X`		
`Left`	`Bust Right`	`X`		
`Right`	`Bust Left`	`X`		
`Right`	`Bust Right`	`X`		
`Right`	`Don't Bust`	`Left`		
`Right`	`Don't Bust`	`X`		

Q4.2 (2 points) If we were to also include the omitted rows, how many rows are in the full table?

Note: Include rows where $T(s, a, s') = 0$.

◯ 8 ◯ 12 ◯ 18 ◯ 24 ◯ 30

Q4.3 (3 points) Suppose that at time step $t = 0$, Pacman believes Blinky is in the **`Left`** square with probability p and is in the **`Right`** square with probability $1 - p$. Recall that Blinky moves left or right, each with probability 0.5, at each time step.

After one time step, with what probability does Pacman believe Blinky is in the **`Left`** square?

◯ 0 ◯ $0.5p$ ◯ p ◯ 0.5 ◯ $1 - p$ ◯ 1

For the remaining subparts, suppose Pacman observes an imperfect sensor to gain information about Blinky's location.

At any given time step t, the sensor's observation (o_t) matches Blinky's location (s_t) with probability 0.7 and does not match Blinky's location with probability 0.3.

Q4.4 (2 points) Fill in the table for $P(O_t \mid S_t)$.

o_t	s_t	$P(o_t \mid s_t)$
`Left`	**`Left`**	
`Left`	**`Right`**	
`Right`	**`Left`**	
`Right`	**`Right`**	

Q4.5 (1 point) Suppose the sensor reports that Blinky is in the **`Left`** square and the discount factor is $\gamma < 1$. What is the optimal action given this observation?

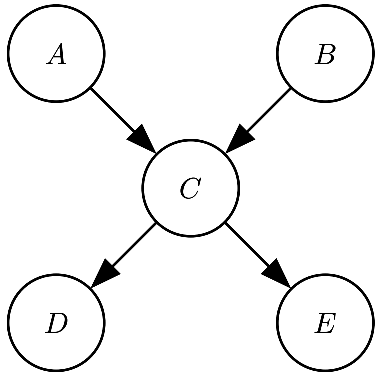
◯ **`Bust Left`** ◯ **`Bust Right`** ◯ **`Don't Bust`**

Q4.6 (2 points) Suppose the sensor reports that Blinky is in the **`Left`** square. Assuming $\gamma < 1$, what is the expected return when acting optimally given this observation?

◯ 0 ◯ 0.3 ◯ 0.4 ◯ 0.5 ◯ 0.7 ◯ 1

Q5 *Costco: Inference and Free Samples* **(15 points)**

Consider the Bayes net below. Each random variable is binary (has two possible values).



Suppose we want to compute $P(B \mid +e)$, using variable elimination.

Q5.1 (1 point) We join and eliminate on A first.

What is the size of the factor generated when we join on A (but before we eliminate on A)?

Note: The size of a factor denotes the number of rows in its CPT.

○ 2^0 ○ 2^1 ○ 2^2 ○ 2^3 ○ 2^4 ○ 2^5

Q5.2 (1 point) We join and eliminate on D next.

What is the size of the factor generated when we join on D (but before we eliminate on D)?

○ 2^0 ○ 2^1 ○ 2^2 ○ 2^3 ○ 2^4 ○ 2^5

Q5.3 (1 point) We join and eliminate on C ~~first~~ next (was clarified on the exam).

What is the size of the factor generated when we join on C (but before we eliminate on C)?

○ 2^0 ○ 2^1 ○ 2^2 ○ 2^3 ○ 2^4 ○ 2^5

Regardless of your previous answers, suppose that the remaining factors after eliminating A, D, and C are $f(B, +e)$ and $P(B)$. Fill in the expression below to derive the desired probability $P(B \mid +e)$.

$$\frac{\text{(i)}}{\sum_{\text{(ii)}} \text{(iii)}}$$

Q5.4 (1 point) Fill in blank (i).

○ $f(B, +e)\ \ P(B)$ ○ $f(B, +e)$ ○ $f(b, +e)\ \ P(b)$ ○ $f(b, +e)$

Q5.5 (1 point) Fill in blank (ii).

○ b ○ b, e ○ e

Q5.6 (2 points) Fill in blank (iii).

○ $f(B, +e)\ P(B)$ ○ $f(B, +e)$ ○ $f(b, +e)\ P(b)$ ○ $f(b, +e)$

(Question 5 continued...)

Now, suppose we want to estimate $P(B \mid +e)$ using sampling.

Q5.7 (2 points) If we use **rejection** sampling, which of these variable orders can be used to produce a valid sample? Select all that apply.

☐ A, B, C, D, E ☐ B, A, C, E, D ☐ C, B, A, E, D

☐ D, E, A, B, C ☐ E, D, C, B, A ◯ None of the above

Q5.8 (2 points) Suppose $P(+e \mid +c) = P(+e \mid -c) = 0.001$, and $P(-e \mid +c) = P(-e \mid -c) = 0.999$.

If we use **likelihood weighting**, what are the weights of the samples produced?

◯ All samples have weight 0.001.

◯ All samples have weight 0.999.

◯ ~~Some~~ Most (was clarified on the exam) samples have weight 0.001, and the remaining samples all have weight 0.999.

◯ ~~Some~~ Most (was clarified on the exam) samples have weight 0.999, and the remaining samples all have weight 0.001.

Q5.9 (2 points) Regardless of your previous answers, consider a scenario where likelihood weighting outputs samples that all have the same weight.

You are given many samples generated from likelihood weighting. Select all strategies that compute a consistent estimate for $P(+b \mid +e)$.

☐ Number of samples with $+b$, divided by total number of samples.

☐ Number of samples with $+e$, divided by total number of samples.

☐ Sum of weights of all samples with $+b$, divided by sum of weights of all samples.

☐ Sum of weights of all samples with $+e$, divided by sum of weights of all samples.

◯ None of the above

Q5.10 (2 points) For this subpart only, suppose we want to estimate $P(+e \mid +a)$ using **rejection** sampling.

We generate samples by sampling each variable, one at a time. Which of the following variables, when sampled first, results in samples being rejected as early as possible?

◯ A ◯ B ◯ C ◯ D ◯ E

Q6 *Scandal* **(11 points)**

Blinky is taking an exam and considers the utility of cheating. We model his thought process as follows:

1. Blinky chooses his level of cheating L, which is a real number from 0 to 1.

   For example, $L = 0$ represents no cheating, $L = 1$ represents the most cheating, and $L = 0.7$ represents a lot of cheating.

2. After the exam, Blinky is either caught ($+c$) or not caught ($-c$). The value of C is sampled from a probability distribution $P(C \mid L)$.
   The probability that Blinky is caught is $P(+c \mid L) = L$, and $P(-c \mid L) = 1 - L$.

3. If Blinky is caught, he receives utility -50. If Blinky is not caught, he receives utility $10 + 100L$.

Q6.1 (1 point) What is Blinky's expected utility if $L = 0$?

○ -50 ○ -40 ○ 5 ○ 10 ○ 60 ○ 100 ○ 110

Q6.2 (1 point) What is Blinky's expected utility if $L = 1$?

○ -50 ○ -40 ○ 5 ○ 10 ○ 60 ○ 100 ○ 110

Q6.3 (2 points) What is Blinky's expected utility if $L = 0.5$?

○ -50 ○ -40 ○ 5 ○ 10 ○ 60 ○ 100 ○ 110

Q6.4 (3 points) What L maximizes Blinky's utility?

Hint: A quadratic $ax^2 + bx + c$ achieves its maximum (assuming one exists) at $x = -\frac{b}{2a}$.

Q6.5 (2 points) Blinky's MEU (maximum expected utility) is 14. What does this number represent?

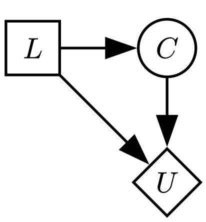
○ If Blinky takes the exam many times, all with the optimal L, he receives utility 14 on average.

○ If Blinky takes the exam many times, all with the optimal L, he always receives utility 14.

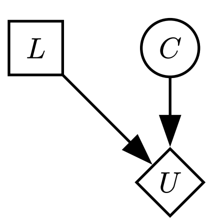
○ When Blinky takes the exam with optimal L, the highest utility he can ever receive is 14.

○ When Blinky takes the exam with optimal L, the lowest utility he can ever receive is 14.

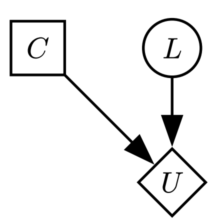
Q6.6 (2 points) Which decision network best models this scenario?



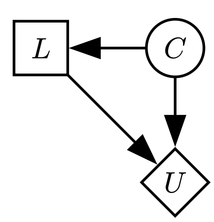
○ Network (i)



○ Network (ii)



○ Network (iii)



○ Network (iv)

Q7 *On the Run* **(8 points)**

Oh no! Blinky was caught cheating on his exam and is now trying to escape! At any given time, Blinky's location (L) is either Dwinelle, VLSB, or Stanford.

We want to compute a belief distribution over Blinky's location using particle filtering.

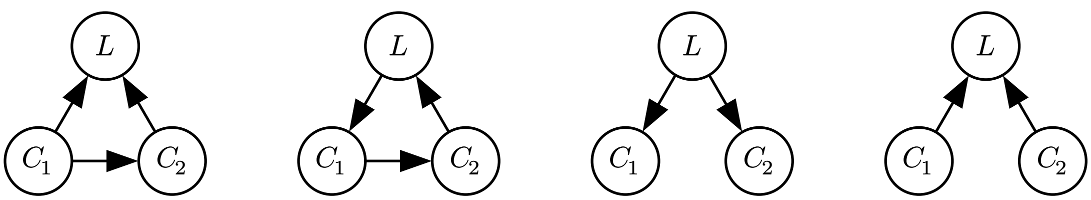
Q7.1 (2 points) Suppose we use 10 particles. We pause the particle filtering algorithm immediately after an observation update.

Which statement implies that $L =$ Stanford with probability 0.9 at this time step?

- ◯ There are exactly 9 particles at Stanford.
- ◯ We add up the un-normalized weights of all the particles at Stanford, and the sum is 9.0.
- ◯ We add up the un-normalized weights of all the particles at Stanford, and the sum is 0.9.
- ◯ We add up the normalized weights of all the particles at Stanford, and the sum is 0.9.

Suppose there are two cameras (C_1, C_2) that are conditionally independent given L. We are only able to observe Blinky's location using these cameras.

Q7.2 (2 points) Which graph always represents the scenario where C_1 and C_2 are conditionally independent given L?



◯ Graph (i) ◯ Graph (ii) ◯ Graph (iii) ◯ Graph (iv)

Q7.3 (2 points) In the observation update, how do we calculate the weight w of a particle?

- ◯ $w \leftarrow P(\ell \mid c_1) \cdot P(\ell \mid c_2)$
- ◯ $w \leftarrow P(c_1 \mid \ell) \cdot P(c_2 \mid \ell)$
- ◯ $w \leftarrow P(c_1, c_2 \mid \ell) \cdot P(\ell)$
- ◯ $w \leftarrow P(c_1 \mid \ell) \cdot P(c_2 \mid c_1)$

Q7.4 (2 points) At some time during the particle filtering algorithm, we have the four weighted particles shown on the right.

During the re-sampling step, what is the probability that a newly re-sampled particle is at Stanford?

◯ 1/8 ◯ 1/4 ◯ 1/2 ◯ 1

	Location (L)	Weight
Particle 1:	Dwinelle	0.4
Particle 2:	Stanford	0.2
Particle 3:	Dwinelle	0.9
Particle 4:	VLSB	0.1

Q8 *Yet Another Machine Learns* **(14 points)**

Q8.1 (2 points) Select all true statements about the Sigmoid function (shown below).

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

☐ Sigmoid outputs values between 0 and 1.

☐ Sigmoid is monotonically increasing.

☐ Sigmoid is linear.

○ None of the above

☐ Sigmoid is non-negative.

For the next two subparts, a perceptron classifier has the following weight vectors for three classes ("Sports", "Music", "Books"):

$$w_{\text{Sports}} = [5, 4, 5] \qquad w_{\text{Music}} = [3, 8, 2] \qquad w_{\text{Books}} = [2, 3, 4]$$

Now, the classifier is updated a single time, using a new data point with feature vector $[1, -1, 2]$ and class "Books".

Q8.2 (2 points) What are the new weights for w_{Music} after the update?

○ $[6, 3, 7]$ ○ $[4, 5, 3]$ ○ $[3, 8, 2]$ ○ $[3, 2, 6]$

○ $[5, 4, 5]$ ○ $[4, 7, 4]$ ○ $[2, 9, 0]$ ○ $[2, 3, 4]$

Q8.3 (2 points) What are the new weights for w_{Books} after the update?

○ $[6, 3, 7]$ ○ $[4, 5, 3]$ ○ $[3, 8, 2]$ ○ $[3, 2, 6]$

○ $[5, 4, 5]$ ○ $[4, 7, 4]$ ○ $[2, 9, 0]$ ○ $[2, 3, 4]$

In the next two subparts, consider a naive Bayes classifier for emails, just like the one from lecture. The classes are "Ham" and "Spam".

Each email is featurized into a 3-element feature vector using the binary bag-of-words model with words "Free", "Money", and "Now". For example, the email "hello money" is featurized into $[0, 1, 0]$, and the email "now now money" is featurized into $[0, 1, 1]$.

Q8.4 (2 points) We want to add a new feature word "Ring" to the classifier. Select all information about the training dataset needed to compute $P(\text{Ring} = 1 \mid \text{class} = \text{Spam})$.

The choices are not independent, e.g. selecting all choices means you need all 4 values. Assume you don't know the total number of Ham and Spam emails.

☐ Number of "Spam" emails containing the word "Ring".

☐ Number of "Ham" emails containing the word "Ring".

☐ Number of "Spam" emails **not** containing the word "Ring".

☐ Number of "Ham" emails **not** containing the word "Ring".

○ None of the above

(Question 8 continued...)

The table below shows the results of running the classifier on a given test dataset:

Predicted Class	True Class	P(Predicted Class | True Class)
Spam	Spam	0.6
Spam	Ham	0.2
Ham	Spam	0.4
Ham	Ham	0.8

In the test dataset, 10% of the data points are "Spam".

Q8.5 (1 point) What is the **precision** of the classifier, treating "Spam" as the positive class?

Hint from lecture slides: $\text{Precision} = \frac{\text{TP}}{\text{TP + FP}}$

○ 0.2 ○ 0.25 ○ 0.4 ○ 0.6 ○ 0.75 ○ 0.8

Q8.6 (1 point) What is the **recall** of the classifier, treating "Spam" as the positive class?

Hint from lecture slides: $\text{Recall} = \frac{\text{TP}}{\text{TP + FN}}$

○ 0.2 ○ 0.25 ○ 0.4 ○ 0.6 ○ 0.75 ○ 0.8

The next two subparts are from the Transformers lecture.

Q8.7 (1 point) True or False: In multi-head attention, each attention head can specialize on extracting different information from the input sequence.

○ True ○ False

Q8.8 (1 point) True or False: Beam search decoding allows LLMs to investigate multiple potential sequences in parallel and select the best one based on a heuristic.

○ True ○ False

The next two subparts are from the guest lectures.

Q8.9 (1 point) Which of these are areas where we see bias in AI? Select all that apply.

☐ Racial ☐ Language ○ None of the above

Q8.10 (1 point) Which of the following is **not** a feature of a good model editing technique?

○ Reliability ○ Incoherence ○ Generalization ○ Locality

Q9 *Did You Pay Attention?* **(10 points)**

Q9.1 (1 point) For unit vectors

$$a = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} \text{ and } b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix},$$

what does the dot product $a \cdot b$ represent?

- ◯ The difference in magnitude between a and b.
- ◯ The similarity between a and b.
- ◯ The sum of a and b.

Q9.2 (2 points) Which of the following is true about Softmax (shown below)?

Note: $\exp(z) = e^z$.

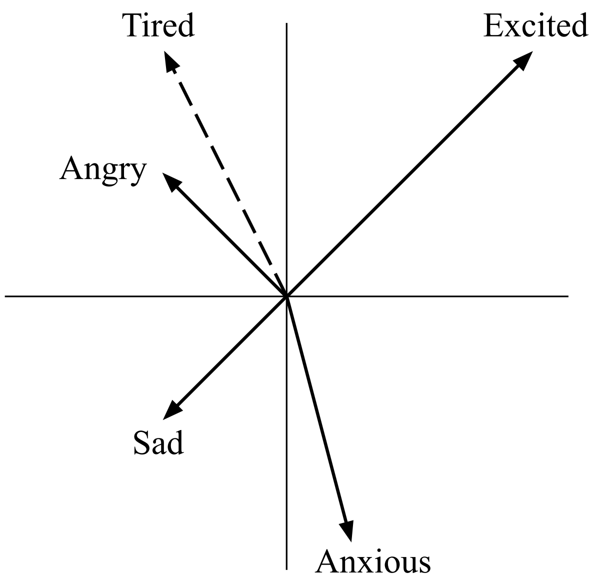
$$\text{Softmax}\left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} \frac{\exp(x_1)}{\sum_{i=1}^{n} \exp(x_i)} \\ \vdots \\ \frac{\exp(x_n)}{\sum_{i=1}^{n} \exp(x_i)} \end{bmatrix}$$

- ☐ The values in the output of Softmax form a valid probability distribution.
- ☐ If $x_i < x_j$, then after applying Softmax, the ith element is less than the jth element.
- ☐ Softmax linearly scales all inputs to between 0 and 1.
- ☐ Softmax can be used to introduce a non-linearity in a neural network.
- ◯ None of the above

This question continues on the next page.

Blinky has a word bank containing {"Angry", "Anxious", "Excited", "Sad"}. He uses a feature extraction function f to convert the words into feature vectors, shown below.

Blinky builds a model that takes in a new word "Tired", featurizes it into $f(\text{Tired})$, and wants to find the most similar word in the word bank.



Words in the bank:

$$f(\text{Angry}) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$f(\text{Anxious}) = \begin{bmatrix} 0.5 \\ -2 \end{bmatrix}$$

$$f(\text{Excited}) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$f(\text{Sad}) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Blinky's new word:

$$f(\text{Tired}) = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

Q9.3 (2 points) Blinky's model uses the following equation, where w is a word in the word bank.

$$\text{SimilarityScore}(\text{Tired}, w) = \frac{\exp(f(\text{Tired}) \cdot f(w))}{\sum_{w' \in \text{bank}} \exp(f(\text{Tired}) \cdot f(w'))}$$

Which w has the highest SimilarityScore with "Tired"?

- ○ Angry
- ○ Anxious
- ○ Excited
- ○ Sad

Q9.4 (2 points) For this subpart only, Xavier creates a XavierSimilarityScore model

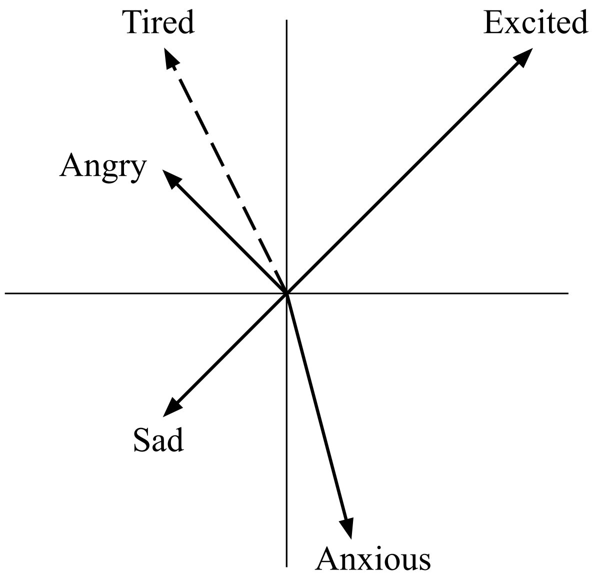
$$\text{XavierSimilarityScore}(\text{Tired}, w) = \frac{\exp(A\, f(\text{Tired}) \cdot f(w))}{\sum_{w' \in \text{bank}} \exp(A\, f(\text{Tired}) \cdot f(w'))}$$

where A is a 2×2 matrix that transforms Tired's feature vector from $f(\text{Tired})$ to $A\, f(\text{Tired})$.

What choice of A causes "Anxious" to have the highest XavierSimilarityScore to "Tired"?

- ○ $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, which leaves $f(\text{Tired})$ unchanged.
- ○ $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, which rotates $f(\text{Tired})$ 90 degrees clockwise.
- ○ $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$, which rotates $f(\text{Tired})$ 180 degrees.
- ○ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, which scales $f(\text{Tired})$ by 2.

The diagram is reprinted for your convenience.



Words in the bank:

$$f(\text{Angry}) = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

$$f(\text{Anxious}) = \begin{bmatrix} 0.5 \\ -2 \end{bmatrix}$$

$$f(\text{Excited}) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$f(\text{Sad}) = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

Blinky's new word:

$$f(\text{Tired}) = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

Q9.5 (2 points) For this subpart only, Noah creates a NoahSimilarityScore model:

$$\text{NoahSimilarityScore}(\text{Tired}, w) = \frac{\exp(f(\text{Tired}) \cdot B\, f(w))}{\sum_{w' \in \text{ bank}} \exp(f(\text{Tired}) \cdot B\, f(w'))}$$

where B is a 2×2 matrix that transforms each bank word's feature vector from $f(w)$ to $B\,f(w)$.

What choice of B causes "Sad" to have the highest NoahSimilarityScore to "Tired"?

- ◯ $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, which leaves $f(w)$ unchanged.
- ◯ $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$, which rotates $f(w)$ 90 degrees clockwise.
- ◯ $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$, which rotates $f(w)$ 180 degrees.
- ◯ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, which scales $f(w)$ by 2.

Q9.6 (1 point) For this subpart only, consider an AttentionScore model:

$$\text{AttentionScore}(\text{Tired}, w) = \frac{\exp(f(\text{Tired}) \cdot f(w))}{\sum_{w' \in \text{ bank}} \exp(f(\text{Tired}) \cdot f(w'))}\, C\, f(w)$$

where C is a 1×2 matrix.

True or False: There exists a choice of C such that "Anxious" has the highest AttentionScore with "Tired" out of all the words in the word bank.

◯ True ◯ False