CS 188 Summer 2016 — Introduction to Artificial Intelligence — Final

- You have approximately 2 hours and 50 minutes.
- The exam is closed book, closed calculator, and closed notes except your two crib sheets.
- Mark your answers ON THE EXAM ITSELF. If you are not sure of your answer you may wish to provide a *brief* explanation or show your work.
- For multiple choice questions,
  - ☐ means mark **all options** that apply
  - ◯ means mark a **single choice**
- There are multiple versions of the exam. For fairness, this does not impact the questions asked, only the ordering of options within a given question.

First name	
Last name	
SID	
edX username	

First and last name of student to your left	
First and last name of student to your right	

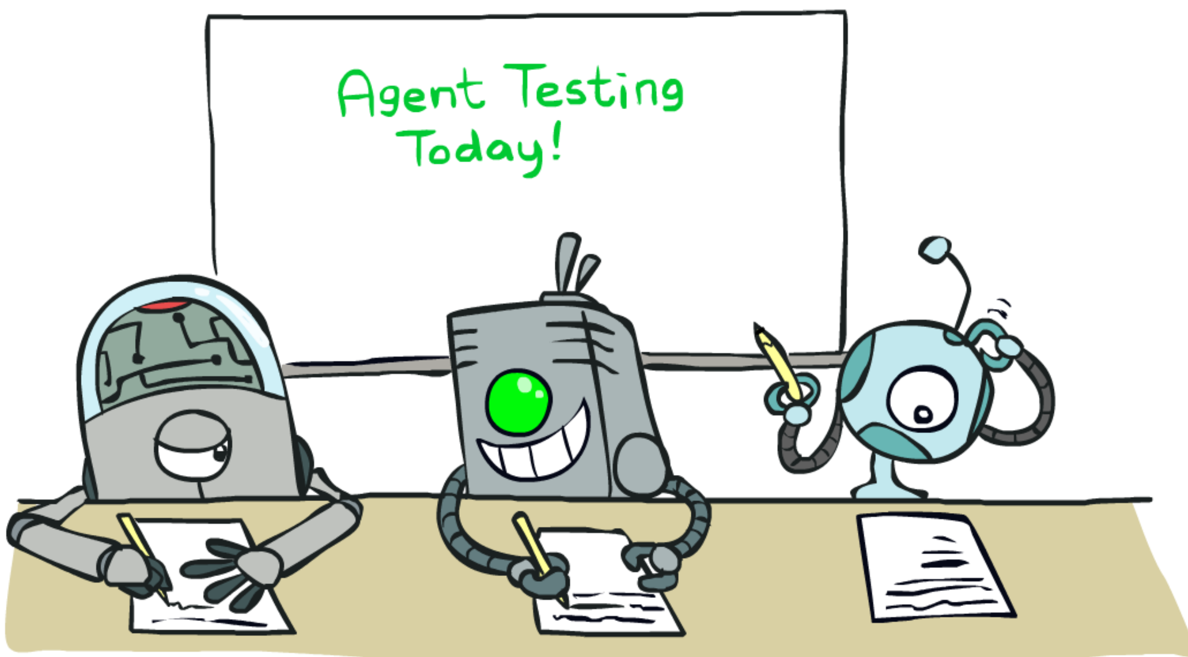
For staff use only:

Q1.	Agent Testing Today!	/1
Q2.	Potpourri	/35
Q3.	GSI Adventures	/18
Q4.	MDPs and RL	/15
Q5.	Perceptron and Kernels	/16
Q6.	Particle Filtering Apprenticeship	/15
Q7.	Neural Network Data Sufficiency	/12
Q8.	Naive Bayes: Pacman or Ghost?	/13
	Total	/125

THIS PAGE IS INTENTIONALLY LEFT BLANK

Q1. [1 pt] Agent Testing Today!

It's testing time! Circle your favorite robot below. We hope you have fun with the rest of the exam!

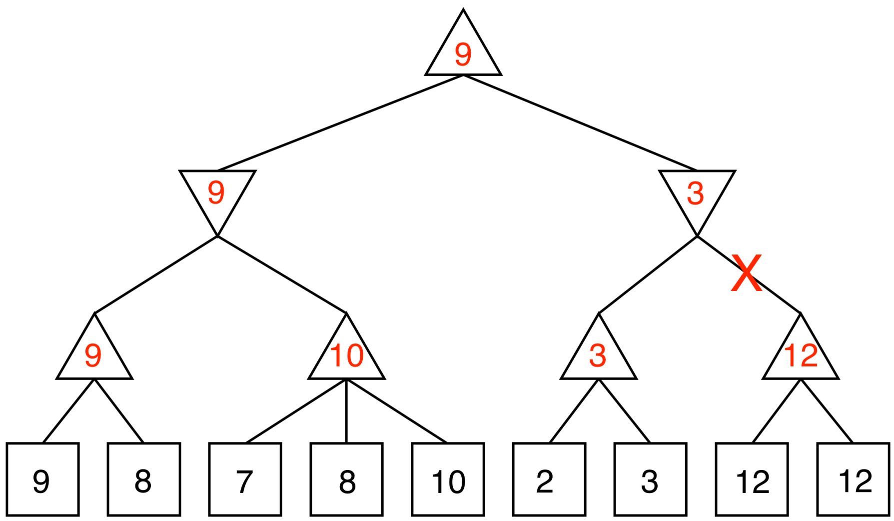


Any answer was acceptable.

Q2. [35 pts] Potpourri

(a) Game trees

(i) [2 pts] Fill in **all** missing values in this game tree. Next, cross out branches pruned by alpha-beta search. (Upward arrows denote a maximizing player, while downward arrows denote a minimizing player.)



(ii) [1 pt] In a **minimax** game, a leaf node that is the first child of its parent may be pruned:

○ Always ○ Sometimes ● Never

(iii) [1 pt] In an **expectimax** game, a leaf node that is the last child of its parent may be pruned:

○ Always ● Sometimes ● Never

We accepted either "Sometimes" (corresponding to a scenario where the values values are known to be bounded) or "Never" (corresponding to a scenario where the values are not known to be bounded).

(b) CSPs

In a general constraint satisfaction problem with N binary-valued variables, backtracking search will backtrack at least __(i)__ times and at most __(ii)__ times. (Choose the tightest upper bound.)

(i) [1 pt] ● $O(1)$ ○ $O(n)$ ○ $O(n^2)$ ○ $O(2^n)$ ○ $O(n!)$

(ii) [1 pt] ○ $O(1)$ ○ $O(n)$ ○ $O(n^2)$ ● $O(2^n)$ ○ $O(n!)$

(c) [1 pt] **Utilities**

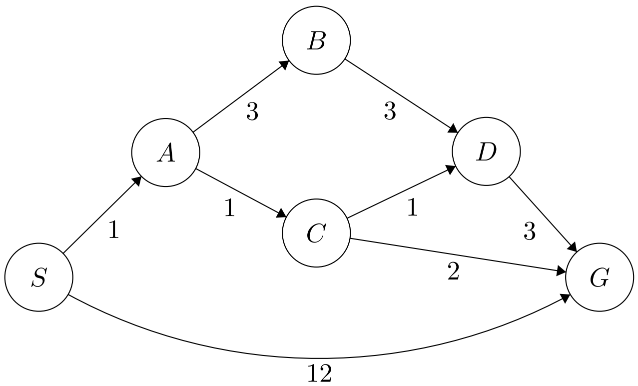
Aldo has a choice between (1) receiving four apples with certainty, and (2) a lottery in which he will receive two, four, or six apples, each with probability 1/3.

Write down a **monotonically decreasing** utility $U(a)$ (where a is the number of apples) such that Aldo strictly prefers to enter the lottery. You may assume $a > 0$.

$U(a) = \frac{1}{a}, -\sqrt{a},$ or any other function with negative first derivative and positive second derivative

(d) Search and Heuristics

Consider the graph and heuristics below for the following problems.



State	$h_1(s)$	$h_2(s)$	$h_3(s)$
S	3	2	2
A	3	2	2
B	5	5	5
C	2	3	2
D	2	2	1
G	0	0	0

For the following, mark all that are true about the heuristic in question.

(i) [1 pt] $h_1(s)$

■ Admissible ■ Consistent □ Neither

(ii) [1 pt] $h_2(s)$

□ Admissible □ Consistent ■ Neither

(iii) [1 pt] $h_3(s)$

■ Admissible □ Consistent □ Neither

For the following search algorithms, fill in the minimal sufficient condition on the heuristic for the algorithm to be guaranteed to be optimal. Fill in "neither" if neither condition is sufficient.

(iv) [1 pt] A^* Tree Search

○ Consistent ● Admissible ○ Neither

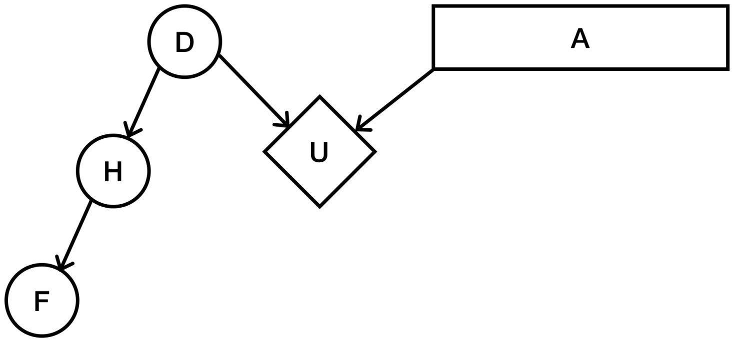
(v) [1 pt] A^* Graph Search

● Consistent ○ Admissible ○ Neither

(vi) [1 pt] Greedy Search

○ Consistent ○ Admissible ● Neither

(e) VPI and Decision Networks For the following question, consider the graph below.



2For the following, decide whether the statement equals 0, does not equal 0, or we need more information to decide. If we need more information to decide, write a relation that would guarantee it to be equal to 0.

(i) [1 pt] $VPI(H)$
- ○ Equal to 0
- ○ Not Equal to 0
- ● Need more information: ____________________

We would need H to be independent of U for this to be true.

(ii) [1 pt] $VPI(H|D)$
- ● Equal to 0
- ○ Not Equal to 0
- ○ Need more information: ____________________

(iii) [1 pt] $VPI(D)$
- ○ Equal to 0
- ○ Not Equal to 0
- ● Need more information: ____________________

We would need D to be independent of U for this to be true.

This question seemed to stump a lot of people, remember, and this is crucial, we can NOT guarantee dependence unless we see the CPTs. Not Equal to 0 should never be the correct answer.

(f) Naive Bayes

(i) [1 pt] In the Naive Bayes model, features are independent effects of the label.

● True ○ False

(ii) [1 pt] Laplace smoothing helps to achieve better accuracy on the training data.

○ True ● False

Consider the following table of data.

A	1	0	1	2	0	1	2	1	2	0
B	0	2	2	1	2	2	1	0	0	1
Y	+	−	+	+	−	+	−	−	−	+

(iii) [1 pt] Find the following quantities. You can leave your answers as fractions.

$P(Y = +) =$ 1/2

$P(A = 0|Y = +) =$ 1/5

$P(B = 2|Y = -) =$ 2/5

(iv) [2 pts] Find the following quantities, using Laplace smoothing with $k = 2$. You can leave your answers as fractions.

$P(A = 1|Y = +) =$ $(3 + 2)/(5 + 2 \cdot 3) = 5/11$

$P(B = 1|Y = -) =$ $(1 + 2)/(5 + 2 \cdot 3) = 3/11$

(g) Perceptron

(i) [1 pt] The perceptron algorithm will converge even if the data is not linearly separable.

○ True ● False

(ii) [1 pt] If while running the perceptron algorithm we make one pass through the data and make no classification mistakes, the algorithm has converged.

● True ○ False

(iii) [1 pt] If we run the perceptron with no bias on d dimensional data, the decision boundary produced by the algorithm is a hyperplane that passes through the origin of $\mathbb{R}^d$.

● True ○ False

(iv) [3 pts] Suppose we have linearly separable three-class data with classes (A, B, C) and run perceptron with initial weights $w_A^{(0)}$, $w_B^{(0)}$, and $w_C^{(0)}$ until convergence. Let N be the number of data points, and let T be the number of updates to the weights before convergence. Let $s = w_A^{(0)} + w_B^{(0)} + w_C^{(0)}$. What is the sum $w_A^{(T)} + w_B^{(T)} + w_C^{(T)}$?

● s
○ Ns
○ Ts
○ NTs
○ Zero vector
○ $\frac{s}{\|s\|}$
○ $N\frac{s}{\|s\|}$
○ $T\frac{s}{\|s\|}$
○ $NT\frac{s}{\|s\|}$
○ None of the above

(v) [2 pts] Consider a perceptron update with step size $\lambda_t = \frac{1}{2^t}$. In other words, for a two class problem the $t-$th iteration is $w_t \leftarrow w_t + \lambda_t y_i x_i$ if (x_i, y_i) is the selected misclassified point to perform the update.

■ This perceptron converges even if the data is not linearly separable.

□ This perceptron update only converges if the data is linearly separable.

■ The order of the data feed will affect the outome of the algorithm.

□ The order of the data feed will not affect the outcome of the algorithm.

(vi) [2 pts] If we run the perceptron update defined above on a linearly separable dataset, it is guaranteed that the algorithm will converge to a linear separator that achieves perfect training accuracy.

○ True ● False

(vii) [2 pts] Aldo wants to use perceptron to build a classifier for his binary class data $\{x_i\}$ with labels $y_i \in \{+1, -1\}$. He has noticed however that his training data is not linearly separable. He has devised a brilliant idea. He decided that he will instead train with the data $z_i = (x_i, y_i)$ and labels y_i. Is the new data, z_i with labels y_i linearly separable?

● Yes ○ No

Comment on Aldo's decision. Do you think it is a good idea? Why or why not?

Although this renders the data linearly separable, it is a very bad idea. The classifier would be unusable, because it would not be able to be evaluated at any new query point. We don't know the right label for this query point!

(h) Optimization

(i) [1 pt] Stochastic gradient descent is guaranteed to arrive at a global optimum.

○ True ● False

(ii) [1 pt] Gradient descent with momentum makes use of second derivative information.

○ True ● False

Q3. [18 pts] GSI Adventures

(a) Missing Exams! The GSIs of 188 are currently looking for where all of the exams have gone! There are 5 GSIs and each one has contact with the other, and they're looking for a grand total of E exams. Imagine Berkeley as an $M \times N$ grid and each GSI starts in a different place. The E exams are spread throughout the Berkeley grid and when a GSI visits a grid space, they are able to pick up all of the exams at that space. During each timestep, a GSI can move 1 grid space. If the exams are not found in T time steps, there will not be time to grade them, and the staff will be forced to give everyone an A. The students know this, so the GSIs must always avoid S students in the grid, otherwise they will steal the exams from them.

(i) [3 pts] Davis and Jacob would like to model this as a search problem. The instructors know where the GSIs start, where the students start, and how they move (that is, student position is a known deterministic function of time). What is a minimal state representation to model this game? Recall that the locations of the exams are not known.

Since we don't know the location of the exams and this is an offline search problem (i.e. we run a simulation of the environment to come up with a full plan before executing it), there is no way we could know how many exams we've picked up. Therefore the only way to model this game as a search problem is to come up with a plan that visits all the squares in T time steps while avoiding the students.

We need to know the position of the GSIs, whether each square has been visited, and the current time step.

(ii) [3 pts] Provide the size of the state representation from above.

$2^{MN}(MN)^5T$

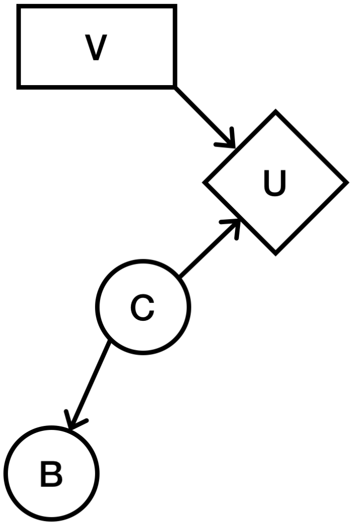
(iii) [2 pts] Which of the following are admissible heuristics for this search problem?

- [ ] The number of exams left to be found
- [ ] The number of exams left to be found divided by 5
- [x] The minimum Manhattan Distance between a GSI and an unvisited grid space
- [ ] The maximum Manhattan Distance between a GSI and an unvisited grid space
- [ ] The number of squares in the grid that have not been visited
- [x] The number of squares in the grid that have not been visited divided by 5

Note that in the correct formulation, we do not have enough information to compute the first two proposed heuristics. In any case, even if we could compute them, neither would be admissible.

(b) The exams have finally been located, and now, it's the students' turn to worry! A student's utility leading up to the exam depends on how hard they study (very hard ($+v$) or just hard ($-v$)) as well as the chance that Davis has a cold around the the exam.

If Davis has a cold ($+c$), he will be too tired to write a hard exam question. He might also be unable to hold office hours, in which case Bob (a reader) will hold office hours instead ($+b$). The decision network and the tables associated with it are shown below:



C	$P(C)$
$+c$	0.5
$-c$	0.5

B	C	$P(B\|C)$
$+b$	$+c$	0.8
$+b$	$-c$	0.1
$-b$	$+c$	0.2
$-b$	$-c$	0.9

V	C	U
$+v$	$+c$	200
$+v$	$-c$	120
$-v$	$+c$	250
$-v$	$-c$	90

Calculate the $VPI(B)$. To do this, in the calculations, calculate $MEU()$, $MEU(+b)$, and $MEU(-b)$. In order to get as much partial credit, provide these calculations, as well as any other calculations necessary, in a neat and readable order. Use the calculated tables below in order to help with the calculations. You may leave your answers as expressions in terms of probabilities in the table and your answers to previous parts.

B	$P(B)$
$+b$	0.45
$-b$	0.55

B	C	$P(C\|B)$
$+b$	$+c$	0.89
$+b$	$-c$	0.11
$-b$	$+c$	0.18
$-b$	$-c$	0.81

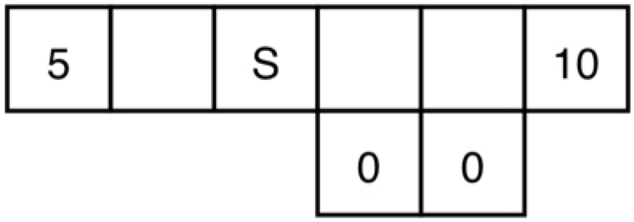
(i) [2 pts] $MEU() = \max(.5 * 200 + .5 * 120, .5 * 250 + .5 * 90) = 170$

(ii) [3 pts] $MEU(+b) = \max(.89 * 200 + .11 * 120, .89 * 250 + .11 * 90) = 232.4$

(iii) [3 pts] $MEU(-b) = \max(.18 * 200 + .81 * 120, .18 * 250 + .81 * 90) = 133.2$

(iv) [2 pts] $VPI(B) = (.45 * MEU(+b) + .55 * MEU(-b)) - MEU()$

Q4. [15 pts] MDPs and RL

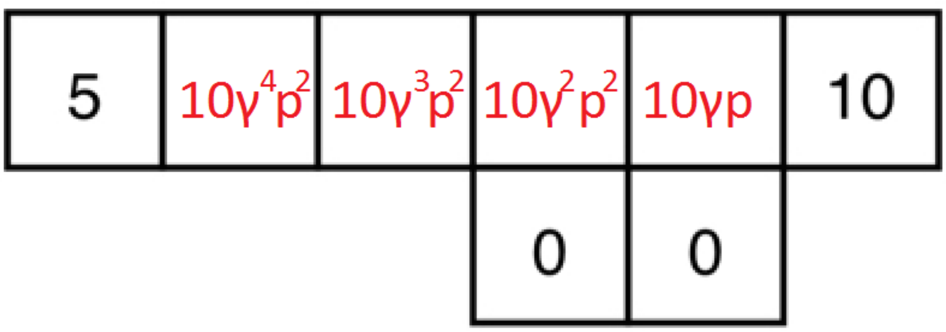


Consider the above gridworld. An agent is currently on grid cell S, and would like to collect the rewards that lie on both sides of it. If the agent is on a numbered square, its only available action is to Exit, and when it exits it gets reward equal to the number on the square. On any other (non-numbered) square, its available actions are to move East and West. Note that North and South are never available actions.

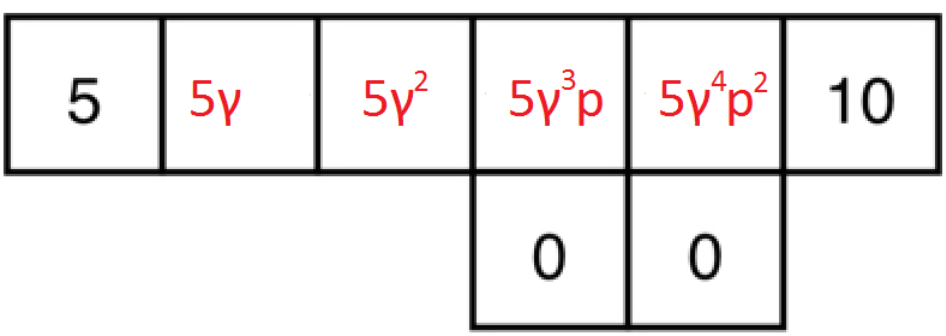
If the agent is in a square with an adjacent square downward, it does not always move successfully: when the agent is in one of these squares and takes a move action, it will only succeed with probability p. With probability $1 - p$, the move action will fail and the agent will instead move downwards. If the agent is not in a square with an adjacent space below, it will always move successfully.

For parts (a) and (b), we are using discount factor $\gamma \in [0, 1]$.

(a) [2 pts] Consider the policy π_{East}, which is to always move East (right) when possible, and to Exit when that is the only available action. For each non-numbered state x in the diagram below, fill in $V^{\pi_{\text{East}}}(x)$ in terms of γ and p.



(b) [2 pts] Consider the policy π_{West}, which is to always move West (left) when possible, and to Exit when that is the only available action. For each non-numbered state x in the diagram below, fill in $V^{\pi_{\text{West}}}(x)$ in terms of γ and p.



(c) [2 pts] For what range of values of p *in terms of* γ is it optimal for the agent to go West (left) from the start state (S)?

We want $5\gamma^2 \geq 10\gamma^3 p^2$, which we can solve to get:

Range: $p \in [0, \frac{1}{\sqrt{2\gamma}}]$

(d) [2 pts] For what range of values of p *in terms of* γ is π_{West} the optimal policy?

We need, for each of the four cells, to have the value of that cell under π_{West} to be at least as large as π_{East}. Intuitively, the farther east we are, the higher the value of moving east, and the lower the value of moving west (since the discount factor penalizes far-away rewards).
Thus, if moving west is the optimal policy, we want to focus our attention on the rightmost cell.
At the rightmost cell, in order for moving west to be optimal, then $V^{\pi_{\text{East}}}(s) \leq V^{\pi_{\text{West}}}(s)$, which is $10\gamma p \leq 5\gamma^4 p^2$, or $p \geq \frac{2}{\gamma^3}$.
However, since γ ranges from 0 to 1, the right side of this expression ranges from 2 to ∞, which means p (a probability, and thus bounded by 1) has no valid value.
Range: $\emptyset$

(e) [2 pts] For what range of values of p *in terms of* γ is π_{East} the optimal policy?

We follow the same logic as in the previous part. Specifically, we focus on the leftmost cell, where the condition for π_{East} to be the optimal policy is: $10\gamma^4 p^2 \geq 5\gamma$, which simplifies to $p \geq \frac{1}{\sqrt{2\gamma^3}}$. Combined with our bound on any probability being in the range $[0, 1]$, we get:

Range: $p \in \left[\frac{1}{\sqrt{2\gamma^3}}, 1\right]$, which could be an empty set depending on γ.

Recall that in approximate Q-learning, the Q-value is a weighted sum of features: $Q(s,a) = \sum_i w_i f_i(s,a)$. To derive a weight update equation, we first defined the loss function $L_2 = \frac{1}{2}(y - \sum_k w_k f_k(x))^2$ and found $dL_2/dw_m = -(y - \sum_k w_k f_k(x)) f_m(x)$. Our label y in this set up is $r + \gamma \max_a Q(s', a')$. Putting this all together, we derived the gradient descent update rule for w_m as $w_m \leftarrow w_m + \alpha\,(r + \gamma \max_a Q(s', a') - Q(s,a))\, f_m(s,a)$.

In the following question, you will derive the gradient descent update rule for w_m using a different loss function:

$$L_1 = \left| y - \sum_k w_k f_k(x) \right|$$

(f) [4 pts] Find dL_1/dw_m. Show work to have a chance at receiving partial credit. Ignore the non-differentiable point.

Note that the derivative of $|x|$ is -1 if $x < 0$ and 1 if $x > 0$. So for L_1, we have:

$$\frac{dL_1}{dw_m} = \begin{cases} -f_m(x) & y - \sum_k w_k f_k(x) > 0 \\ f_m(x) & y - \sum_k w_k f_k(x) < 0 \end{cases}$$

(g) [1 pt] Write the gradient descent update rule for w_m, using the L_1 loss function.

$$\begin{aligned} w_m &\leftarrow w_m - \alpha dL_1/dw_m \\ &\leftarrow \begin{cases} w_m + \alpha f_m(x) & y - \sum_k w_k f_k(x) > 0 \\ w_m - \alpha f_m(x) & y - \sum_k w_k f_k(x) < 0 \end{cases} \end{aligned}$$

Q5. [16 pts] Perceptron and Kernels

A kernel is a mapping $K(x, y)$ from pairs vectors in $\mathbb{R}^d$ into the real numbers such that $K(x, y) = \Phi(x) \cdot \Phi(y)$ where Φ is a mapping from $\mathbb{R}^d$ into $\mathbb{R}^D$ where D is possibly different from d and even infinite. We say that a mapping $K(x, y)$ for which such Φ exists is a valid kernel.

(a) The following binary class data has two features, A and B.

Index	A	B	Class
1.	1	1	1
2.	0	3	-1
3.	1	-1	1
4.	3	0	-1
5.	-1	1	1
6.	0	-3	-1
7.	-1	-1	1
8.	-3	0	-1

(i) [3 pts] Select all true statements:

- ☐ This data is linearly separable.
- ■ This data is linearly separable if we use a feature map $\phi((A, B)) = (A^2, B^2, 1)$.
- ■ There exists a kernel such that this data is linearly separable.
- ■ For all datasets in which no data point is labeled in more than one distinct way, there exists a kernel such that the data is linearly separable.
- ☐ For all datasets, there exists a kernel such that the data is linearly separable.
- ☐ For all valid kernels, there exists a dataset with at least one point from each class that is linearly separable under that kernel.
- ☐ None of the above.

We will be running both the primal (normal) binary (not multiclass) perceptron and dual binary perceptron algorithms on this dataset. We will initialize the weight vector w to $(1, 1)$ for the primal perceptron algorithm. Accordingly, we will initialize the α vector to $(1, 0, 0, 0, 0, 0, 0, 0)$ for the dual perceptron algorithm with the kernel $K(x, y) = x \cdot y$. Pass through the data using the indexing order provided. There is no bias term.

Write your answer in the box provided. Show your work outside of the boxes to have a chance at receiving partial credit.

(ii) [1 pt]
What is the first misclassified point?

Point 2.

(iii) [1 pt] For the *primal* perceptron algorithm, what is the weight vector after the first weight update?

The weight vector after the first weight update will be:

$$w = (1, 1) - (0, 3) = (1, -2) \quad (1)$$

For your convenience, the data is duplicated on this page.

Index	A	B	Class
1.	1	1	1
2.	0	3	-1
3.	1	-1	1
4.	3	0	-1
5.	-1	1	1
6.	0	-3	-1
7.	-1	-1	1
8.	-3	0	-1

(iv) [1 pt] For the *dual* perceptron algorithm, what is the α vector after the first weight update?

The α vector after the first update will be:

$$\alpha = (1, -1, 0, 0, 0, 0, 0, 0) \tag{2}$$

(v) [1 pt] What is the second misclassified point?

Point 4.

(vi) [1 pt] For the *primal* perceptron algorithm, what is the weight vector after the second weight update?

The weights after the second weight update will be:

$$w = (1, -2) - (3, 0) = (-2, -2) \tag{3}$$

(vii) [1 pt] For the *dual* perceptron algorithm, what is the α vector after the second weight update?

The α vector after the second update will be:

$$\alpha = (1, -1, 0, -1, 0, 0, 0, 0) \tag{4}$$

(b) [3 pts] Consider the following kernel function: $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^2$ where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$. Find a valid Φ map for this kernel. That is, find a vector-to-vector function ϕ such that $\phi(\mathbf{x}) \cdot \phi(\mathbf{y}) = K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^2$. Show work to have a chance at receiving partial credit. Any precise answer format is acceptable.

Expanding $(x \cdot y)^2 = (x_1 y_1 + x_2 y_2)^2 = x_1^2 y_1^2 + 2 x_1 y_1 x_2 y_2 + x_2^2 y_2^2$ so the mapping $\Phi(x) = [x_1^2, \sqrt{2} x_1 x_2, x_2^2]$ is valid.

(c) We have n data points, $\{(x_i, y_i)\}_{i=1}^n$, with $x_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, \ldots, M\}$. That is, they are labelled as belonging to one of M classes. We will run the multiclass perceptron algorithm with an RBF kernel:

$$K(x_i, x_j) = \exp(- \parallel x_i - x_j \parallel^2) \tag{5}$$

Denote the dual weights at time t as $\alpha_y^{(t)} = (\alpha_{y,1}^{(t)}, \cdots, \alpha_{y,K}^{(t)})$ for all classes $y = 1, \cdots, M$.

(i) [1 pt] What is the right value for K, the dimension of each of the dual weight vectors?

- ● n
- ○ M
- ○ Mn
- ○ $M + n$

(ii) [3 pts]

Assume that for some t, and for all y, $\alpha_y^{(t)}$ has only one nonzero entry. This single nonzero entry equals one. All the nonzero entries occur at different indices for different y. Describe the decision regions in $\mathcal{R}^d$ for the M classes in terms of distances between points.

The nonzero entries of $\alpha_y^{(t)}$ correspond to M points in the training data. Call these points the class centers. Each of these will also correspond to some class. Not necessarily the training point class.

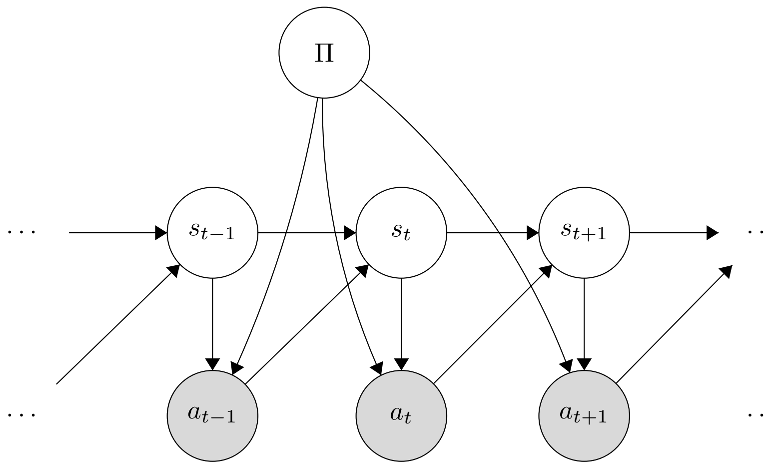
Any new query point $x \in \mathbb{R}^d$ will be labeled as the class corresponding to the closest class center.

Q6. [15 pts] Particle Filtering Apprenticeship

Consider a modified version of the apprenticeship problem. We are observing an agent's actions in an MDP and are trying to determine which out of a set $\{\pi_1, \ldots, \pi_n\}$ the agent is following. Let the random variable Π take values in that set and represent the policy that the agent is acting under. We consider only *stochastic* policies, so that A_t is a random variable with a distribution conditioned on S_t and Π. As in a typical MDP, S_t is a random variable with a distribution conditioned on S_{t-1} and A_{t-1}. The full Bayes net is shown below.

The agent acting in the environment knows what state it is currently in (as is typical in the MDP setting). Unfortunately, however, we, the observer, cannot see the states S_t. Thus we are forced to use an adapted particle filtering algorithm to solve this problem. Concretely, we will develop an efficient algorithm to estimate $P(\Pi \mid a_{1:t})$.

(a) The Bayes net for part (a) is



(i) [3 pts] Select all of the following that are guaranteed to be true in this model for $t > 10$:

- ☐ $S_t \perp\!\!\!\perp S_{t-2} \mid S_{t-1}$
- ■ $S_t \perp\!\!\!\perp S_{t-2} \mid S_{t-1}, A_{1:t-1}$
- ☐ $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi$
- ☐ $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, A_{1:t-1}$
- ■ $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, S_{t-1}$
- ■ $S_t \perp\!\!\!\perp S_{t-2} \mid \Pi, S_{t-1}, A_{1:t-1}$
- ☐ None of the above

We will compute our estimate for $P(\Pi \mid a_{1:t})$ by coming up with a recursive algorithm for computing $P(\Pi, S_t \mid a_{1:t})$. (We can then sum out S_t to get the desired distribution; in this problem we ignore that step.)

(ii) [2 pts] Write a recursive expression for $P(\Pi, S_t \mid a_{1:t})$ in terms of the CPTs in the Bayes net above. Hint: Think of the forward algorithm.

$$P(\Pi, S_t \mid a_{1:t}) \propto \sum_{s_{t-1}} P(\Pi, s_{t-1} \mid a_{1:t-1}) P(a_t \mid S_t, \Pi) P(S_t \mid s_{t-1}, a_{t-1})$$

We now try to adapt particle filtering to approximate this value. Each particle will contain a single state s_t and a potential policy π_i.

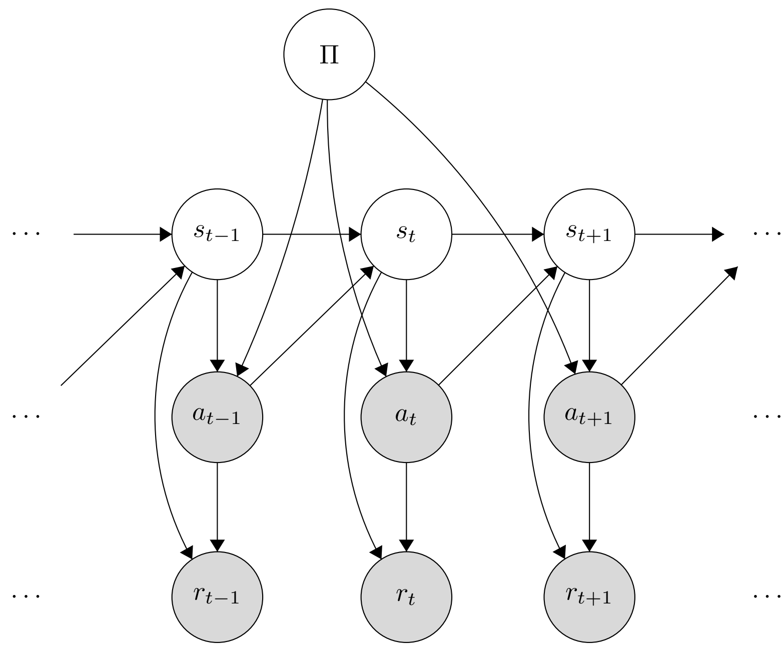
(iii) [2 pts] The following is pseudocode for the body of the loop in our adapted particle filtering algorithm. Fill in the boxes with the correct values so that the algorithm will approximate $P(\Pi, S_t \mid a_{1:t})$.

1. Elapse time: for each particle (s_t, π_i), sample a successor s_{t+1} from $P(S_{t+1} \mid s_t, a_t)$. The policy π' in the new particle is π_i.

2. Incorporate evidence: To each new particle (s_{t+1}, π'), assign weight $P(a_{t+1} \mid s_{t+1}, \pi')$.

3. Resample particles from the weighted particle distribution.

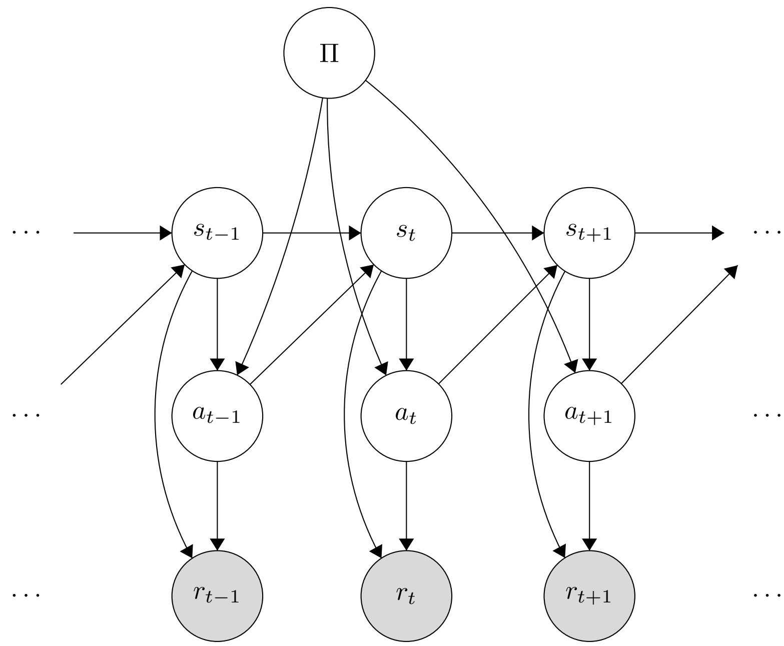
(b) [1 pt] We now observe the acting agent's actions *and* rewards at each time step (but we still don't know the states). Unlike the MDPs in lecture, here we use a stochastic reward function, so that R_t is a random variable with a distribution conditioned on S_t and A_t. The new Bayes net is given by



Notice that the observed rewards do in fact give useful information since d-separation does not give that $R_t \perp\!\!\!\perp \Pi \mid A_{1:t}$. Give an active path connecting R_t and Π when $A_{1:t}$ are observed. Your answer should be an ordered list of nodes in the graph, for example "$S_t, S_{t+1}, A_t, \Pi, A_{t-1}, R_{t-1}$".

R_t, S_t, A_t, Π. This list reversed is also correct, and many other similar (though more complicated) paths are also correct.

(c) We now observe *only* the sequence of rewards and no longer observe the sequence of actions. The new Bayes net is:



We will compute our estimate for $P(\Pi \mid r_{1:t})$ by coming up with a recursive algorithm for computing $P(\Pi, S_t, A_t \mid r_{1:t})$. (We can then sum out S_t and A_t to get the desired distribution; in this problem we ignore that step.)

(i) [2 pts] Write a recursive expression for $P(\Pi, S_t, A_t \mid r_{1:t})$ in terms of the CPTs in the Bayes net above.

$$P(\Pi, S_t, A_t \mid r_{1:t}) \propto \sum_{s_{t-1}} \sum_{a_{t-1}} P(\Pi, s_{t-1}, a_{t-1} \mid r_{1:t-1}) P(A_t \mid S_t, \Pi) P(S_t \mid s_{t-1}, a_{t-1}) P(r_t \mid S_t, A_t)$$

We now try to adapt particle filtering to approximate this value. Each particle will contain a single state s_t, a single action a_t, and a potential policy π_i.

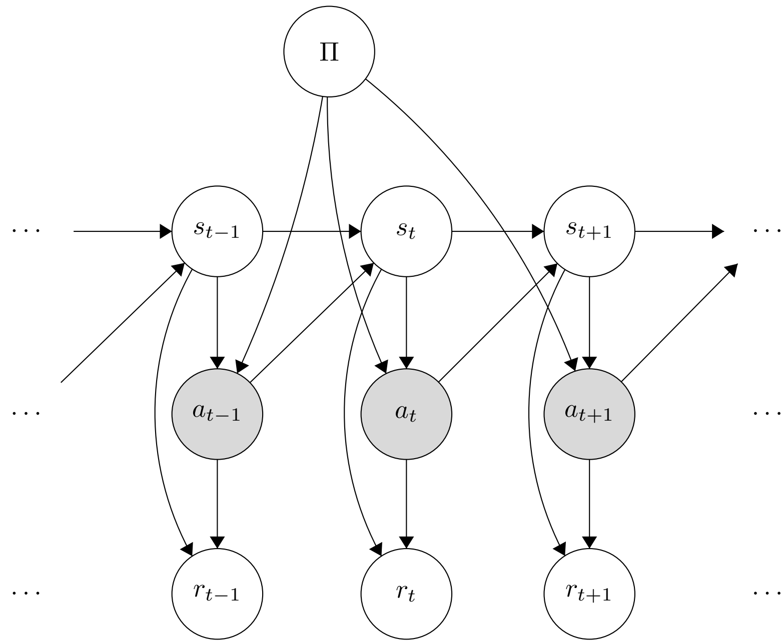
(ii) [2 pts] The following is pseudocode for the body of the loop in our adapted particle filtering algorithm. Fill in the boxes with the correct values so that the algorithm will approximate $P(\Pi, S_t, A_t \mid r_{1:t})$.

1. Elapse time: for each particle (s_t, a_t, π_i), sample a successor state s_{t+1} from [$P(S_{t+1} \mid s_t, a_t)$]. Then, sample a successor action a_{t+1} from [$P(A_{t+1} \mid s_{t+1}, \pi_i)$]. The policy π' in the new particle is [π_i].

2. Incorporate evidence: To each new particle (s_{t+1}, a_{t+1}, π'), assign weight [$P(r_{t+1} \mid s_{t+1}, a_{t+1})$].

3. Resample particles from the weighted particle distribution.

(d) Finally, consider the following Bayes net:



Here, the task is identical to that in part (a); we see only the actions and want to approximate $P(\Pi, \mid a_{1:t})$. However, now we are also accounting for the hidden reward variables.

(i) [1 pt] For a fixed state action pair (s_t, a_t), what is $\sum_{r_t} P(r_t \mid s_t, a_t)$?

1

Suppose for the following questions we adapt particle filtering to this model as in previous parts. In particular, in this algorithm, our particles will also track r_t values.

(ii) [1 pt] Comparing to the algorithm in (a), with the same number of particles, this algorithm will give an estimate of $P(\Pi \mid a_{1:t})$ that is

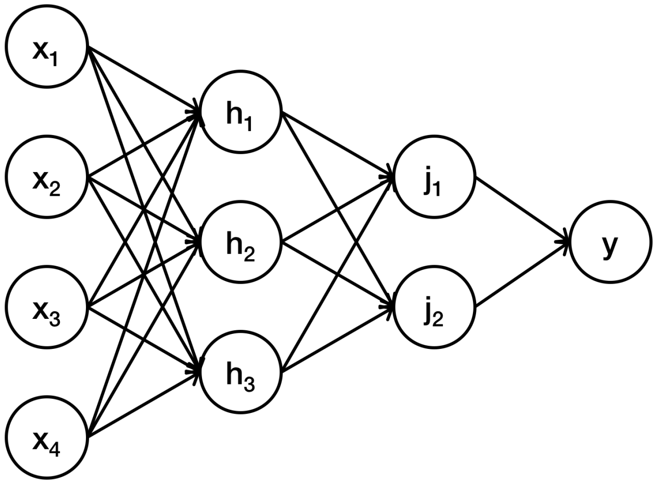
○ More accurate ● Equally accurate ○ Less accurate

(iii) [1 pt] Comparing to the algorithm in (a), with the same number of particles, to compute an estimate, this algorithm will take

● More time ○ The same amount of time ○ Less time

Q7. [12 pts] Neural Network Data Sufficiency

The next few problems use the below neural network as a reference. Neurons h_{1-3} and j_{1-2} all use ReLU activation functions. Neuron y uses the identity activation function: $f(x) = x$. In the questions below, let $w_{a,b}$ denote the weight that connects neurons a and b. Also, let o_a denote the value that neuron a outputs to its next layer.



Given this network, in the following few problems, you have to decide whether the data given are sufficient for answering the question.

(a) [2 pts] Given the above neural network, what is the value of o_y?

Data item 1: the values of all weights in the network and the values o_{h_1}, o_{h_2}, o_{h_3}
Data item 2: the values of all weights in the network and the values o_{j_1}, o_{j_2}

- ○ Data item (1) alone is sufficient, but data item (2) alone is not sufficient to answer the question.
- ○ Data item (2) alone is sufficient, but data item (1) alone is not sufficient to answer the question.
- ○ Both statements taken together are sufficient, but neither data item alone is sufficient.
- ● Each data item alone is sufficient to answer the question.
- ○ Statements (1) and (2) together are not sufficient, and additional data is needed to answer the question.

(b) [2 pts] Given the above neural network, what is the value of o_{h_1}?

Data item 1: the neuron input values, i.e., o_{x_1} through o_{x_4}
Data item 2: the values o_{j_1}, o_{j_2}

- ○ Data item (1) alone is sufficient, but data item (2) alone is not sufficient to answer the question.
- ○ Data item (2) alone is sufficient, but data item (1) alone is not sufficient to answer the question.
- ○ Both statements taken together are sufficient, but neither data item alone is sufficient.
- ○ Each data item alone is sufficient to answer the question.
- ● Statements (1) and (2) together are not sufficient, and additional data is needed to answer the question.

(c) [2 pts] Given the above neural network, what is the value of o_{j_1}?

Data item 1: the values of all weights connecting neurons h_1, h_2, h_3 to j_1, j_2
Data item 2: the values o_{h_1}, o_{h_2}, o_{h_3}

- ○ Data item (1) alone is sufficient, but data item (2) alone is not sufficient to answer the question.
- ○ Data item (2) alone is sufficient, but data item (1) alone is not sufficient to answer the question.
- ● Both statements taken together are sufficient, but neither data item alone is sufficient.
- ○ Each data item alone is sufficient to answer the question.
- ○ Statements (1) and (2) together are not sufficient, and additional data is needed to answer the question.

(d) [2 pts] Given the above neural network, what is the value of $\partial o_y / \partial w_{j_2,y}$?

Data item 1: the value of o_{j_2}
Data item 2: all weights in the network and the neuron input values, i.e., o_{x_1} through o_{x_4}

- ○ Data item (1) alone is sufficient, but data item (2) alone is not sufficient to answer the question.
- ○ Data item (2) alone is sufficient, but data item (1) alone is not sufficient to answer the question.
- ○ Both statements taken together are sufficient, but neither data item alone is sufficient.
- ● Each data item alone is sufficient to answer the question.
- ○ Statements (1) and (2) together are not sufficient, and additional data is needed to answer the question.

(e) [2 pts] Given the above neural network, what is the value of $\partial o_y / \partial w_{h_2,j_2}$?

Data item 1: the value of $w_{j_2,y}$
Data item 2: the value of $\partial o_{j_2} / \partial w_{h_2,j_2}$

- ○ Data item (1) alone is sufficient, but data item (2) alone is not sufficient to answer the question.
- ○ Data item (2) alone is sufficient, but data item (1) alone is not sufficient to answer the question.
- ● Both statements taken together are sufficient, but neither data item alone is sufficient.
- ○ Each data item alone is sufficient to answer the question.
- ○ Statements (1) and (2) together are not sufficient, and additional data is needed to answer the question.

(f) [2 pts] Given the above neural network, what is the value of $\partial o_y / \partial w_{x_1,h_3}$?

Data item 1: the value of all weights in the network and the neuron input values, i.e., o_{x_1} through o_{x_4}
Data item 2: the value of w_{x_1,h_3}

- ● Data item (1) alone is sufficient, but data item (2) alone is not sufficient to answer the question.
- ○ Data item (2) alone is sufficient, but data item (1) alone is not sufficient to answer the question.
- ○ Both statements taken together are sufficient, but neither data item alone is sufficient.
- ○ Each data item alone is sufficient to answer the question.
- ○ Statements (1) and (2) together are not sufficient, and additional data is needed to answer the question.

Q8. [13 pts] Naive Bayes: Pacman or Ghost?

You are standing by an exit as either Pacmen or ghosts come out of it. Every time someone comes out, you get two observations: a visual one and an auditory one, denoted by the random variables X_v and X_a, respectively. The visual observation informs you that the individual is either a Pacman ($X_v = 1$) or a ghost ($X_v = 0$). The auditory observation X_a is defined analogously. Your observations are a noisy measurement of the individual's true type, which is denoted by Y. After the indiviual comes out, you find out what they really are: either a Pacman ($Y = 1$) or a ghost ($Y = 0$). You have logged your observations and the true types of the first 20 individuals:

individual i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
first observation $X_v^{(i)}$	0	0	1	0	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	0
second observation $X_a^{(i)}$	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
individual's type $Y^{(i)}$	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

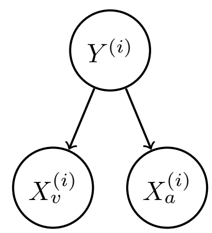
The superscript (i) denotes that the datum is the ith one. Now, the individual with $i = 20$ comes out, and you want to predict the individual's type $Y^{(20)}$ given that you observed $X_v^{(20)} = 1$ and $X_a^{(20)} = 1$.

(a) Assume that the types are independent, and that the observations are independent conditioned on the type. You can model this using naïve Bayes, with $X_v^{(i)}$ and $X_a^{(i)}$ as the features and $Y^{(i)}$ as the labels. Assume the probability distributions take on the following form:

$$P(X_v^{(i)} = x_v | Y^{(i)} = y) = \begin{cases} p_v & \text{if } x_v = y \\ 1 - p_v & \text{if } x_v \neq y \end{cases}$$

$$P(X_a^{(i)} = x_a | Y^{(i)} = y) = \begin{cases} p_a & \text{if } x_a = y \\ 1 - p_a & \text{if } x_a \neq y \end{cases}$$

$$P(Y^{(i)} = 1) = q$$



for $p_v, p_a, q \in [0, 1]$ and $i \in \mathbb{N}$.

(i) [3 pts] What's the maximum likelihood estimate of p_v, p_a and q?

$p_v =$ $\frac{4}{5}$ $\quad p_a =$ $\frac{3}{5}$ $\quad q =$ $\frac{1}{2}$

To estimate q, we count 10 $Y = 1$ and 10 $Y = 0$ in the data. For p_v, we have $p_v = 8/10$ cases where $X_v = 1$ given $Y = 1$ and $1 - p_v = 2/10$ cases where $X_v = 1$ given $Y = 0$. So $p_v = 4/5$. For p_a, we have $p_a = 2/10$ cases where $X_a = 1$ given $Y = 1$ and $1 - p_v = 0/10$ cases where $X_v = 1$ given $Y = 0$. The average of 2/10 and 1 is 3/5.

(ii) [3 pts] What is the probability that the next individual is Pacman given your observations? Express your answer in terms of the parameters p_v, p_a and q (you might not need all of them).

$P(Y^{(20)} = 1 | X_v^{(20)} = 1, X_a^{(20)} = 1) =$ $\frac{p_v p_a q}{p_v p_a q + (1-p_v)(1-p_a)(1-q)}$

The joint distribution $P(Y = 1, X_v = 1, X_a = 1) = p_v p_a q$. For the denominator, we need to sum out over Y, that is, we need $P(Y = 1, X_v = 1, X_a = 1) + P(Y = 0, X_v = 1, X_a = 1)$.

Now, assume that you are given additional information: you are told that the individuals are actually coming out of a bus that just arrived, and each bus carries *exactly* 9 individuals. Unlike before, the types of every 9 consecutive individuals are *conditionally* independent given the bus type, which is denoted by Z. Only after all of the 9 individuals have walked out, you find out the bus type: one that carries mostly Pacmans ($Z = 1$) or one that carries mostly ghosts ($Z = 0$). Thus, you only know the bus type in which the first 18 individuals came in:

individual i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
first observation $X_v^{(i)}$	0	0	1	0	1	0	0	1	1	1	0	1	1	0	1	1	1	0	0	0
second observation $X_a^{(i)}$	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
individual's type $Y^{(i)}$	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0

bus j	0	1	
bus type $Z^{(j)}$	0	1	

(b) You can model this using a variant of naïve bayes, where now 9 consecutive labels $Y^{(i)}, \dots, Y^{(i+8)}$ are *conditionally* independent given the bus type $Z^{(j)}$, for bus j and individual $i = 9j$. Assume the probability distributions take on the following form:

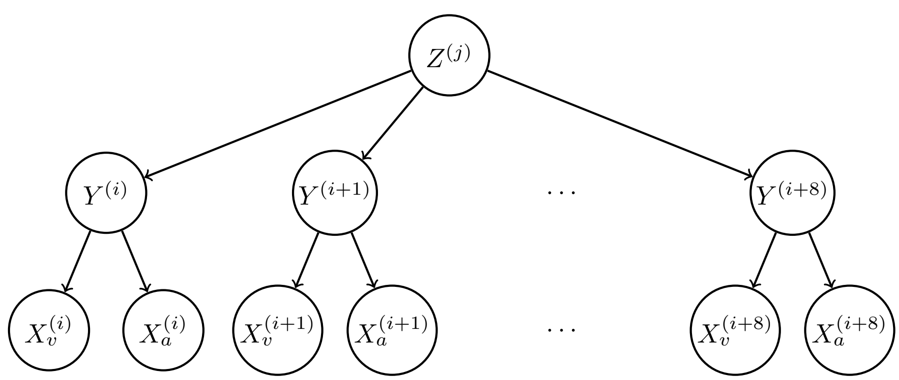
$$P(X_v^{(i)} = x_v | Y^{(i)} = y) = \begin{cases} p_v & \text{if } x_v = y \\ 1 - p_v & \text{if } x_v \neq y \end{cases}$$

$$P(X_a^{(i)} = x_a | Y^{(i)} = y) = \begin{cases} p_a & \text{if } x_a = y \\ 1 - p_a & \text{if } x_a \neq y \end{cases}$$

$$P(Y^{(i)} = 1 | Z^{(j)} = z) = \begin{cases} q_0 & \text{if } z = 0 \\ q_1 & \text{if } z = 1 \end{cases}$$

$$P(Z^{(j)} = 1) = r$$

for $p, q_0, q_1, r \in [0, 1]$ and $i, j \in \mathbb{N}$.



(i) [3 pts] What's the maximum likelihood estimate of q_0, q_1 and r?

$q_0 =$ $\frac{2}{9}$ $\qquad q_1 =$ $\frac{8}{9}$ $\qquad r =$ $\frac{1}{2}$

For r, we've seen one ghost bus and one pacman bus, so $r = 1/2$. For q_0, we're finding $P(Y = 1|Z = 0)$, which is $2/9$. For q_1, we're finding $P(Y = 1|Z = 1)$, which is $8/9$.

(ii) [4 pts] Compute the following joint probability. Simplify your answer as much as possible and express it in terms of the parameters p_v, p_a, q_0, q_1 and r (you might not need all of them).

$P(Y^{(20)} = 1, X_v^{(20)} = 1, X_a^{(20)} = 1, Y^{(19)} = 1, Y^{(18)} = 1) =$ $p_a p_v [q_0^3(1-r) + q_1^3 r]$

$$
\begin{aligned}
&P(Y^{(20)} = 1, X_v^{(20)} = 1, X_a^{(20)} = 1, Y^{(19)} = 1, Y^{(18)} = 1) \\
&= \sum_z P(Y^{(20)} = 1 | Z^{(2)} = z) P(Z^{(2)} = z) P(X_v^{(20)} = 1 | Y^{(20)} = 1) P(X_a^{(20)} = 1 | Y^{(20)} = 1) \\
&\qquad P(Y^{(19)} = 1 | Z^{(2)} = z) P(Y^{(18)} = 1 | Z^{(2)} = z) \\
&= q_0(1-r) p_a p_v q_0 q_0 + q_1 r p_a p_v q_1 q_1 \\
&= p_a p_v [q_0^3(1-r) + q_1^3 r]
\end{aligned}
$$

THIS PAGE IS INTENTIONALLY LEFT BLANK

THIS PAGE IS INTENTIONALLY LEFT BLANK