CS221 Exam Solutions

CS221
November 28, 2017

Name: ____________________
by writing my name I agree to abide by the honor code

SUNet ID: ____________________

Read all of the following information before starting the exam:

- This test has 3 problems and is worth 150 points total. It is your responsibility to make sure that you have all of the pages.
- Keep your answers precise and concise. Show all work, clearly and in order, or else points will be deducted, even if your final answer is correct.
- Don't spend too much time on one problem. Read through all the problems carefully and do the easy ones first. Try to understand the problems intuitively; it really helps to draw a picture.
- You cannot use any external aids except one double-sided $8\frac{1}{2}$" x 11" page of notes.
- Good luck!

Problem	Part	Max Score	Score
1	a	15	
	b	15	
	c	20	
2	a	10	
	b	10	
	c	10	
	d	10	
	e	10	
3	a	15	
	b	10	
	c	15	
	d	10	

Total Score: $\square + \square + \square = \square$

1. Learning (*50 points*)

a. (*15 points*) **[Generalization]**

For problems (i)–(iii), circle one of the bolded options.

(i) [3 points] To decrease training error, would you want **more** or **less** data?

Solution Less. Fewer data points are easier to fit.

(ii) [3 points] To decrease training error, would you want to **add** or **remove** features?

Solution Add. More features makes it easier to fit the data.

(iii) [3 points] To decrease training error, would you want to make the set of hypotheses **smaller** or **larger**?

Solution Larger. More hypotheses makes it easier to fit the data.

(iv) [3 points] If a learning algorithm generalizes very well, what does this say about the training error and the test error? Your answer should be one sentence.

Solution The training and the test errors are approximately equal. Note that strictly speaking, we do not require the test error to be small.

(v) [3 points] In class, we talked about dividing the data into train, validation, and test sets. Give a way that you might use the validation set. Your answer should be one sentence.

Solution The validation set can be used to (1) tune hyperparameters, (2) choose features, (3) choose hypothesis class, or other decisions where you want to estimate the test error, without using the test error.

b. (*15 points*) **[Clustering]**

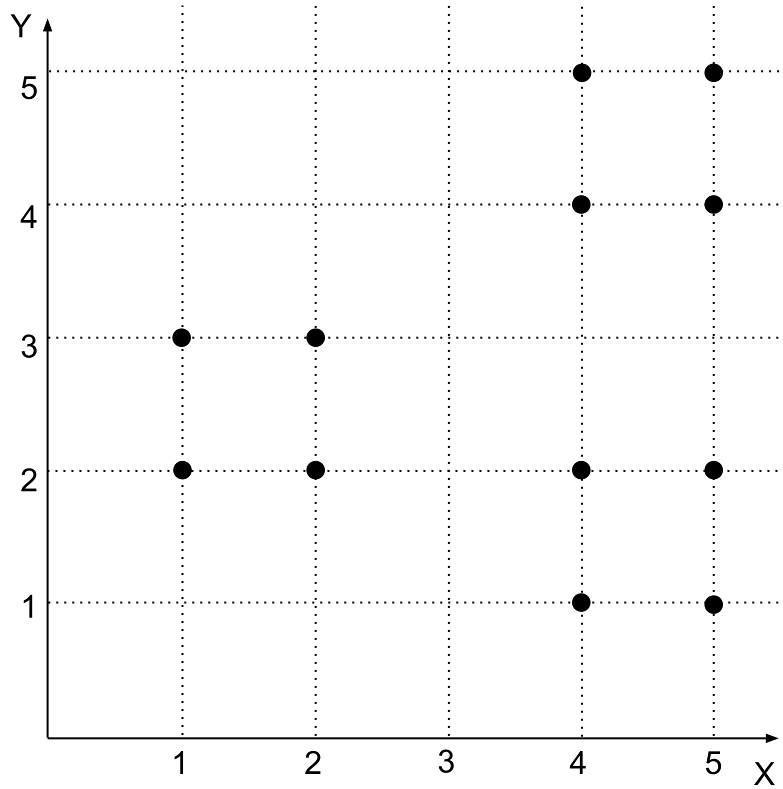

Figure 1: Points to be clustered.

Suppose we have 12 points shown in Figure 1. Recall that the k-means algorithm tries to minimize the reconstruction loss, alternating between optimizing over the cluster centroids and optimizing over the cluster assignments. When we optimize over the assignments, suppose we **tie break assigning points to the cluster with the lower index** (e.g. points to centroid μ_1 rather than centroid μ_2 if the distance to both centroids are equal).

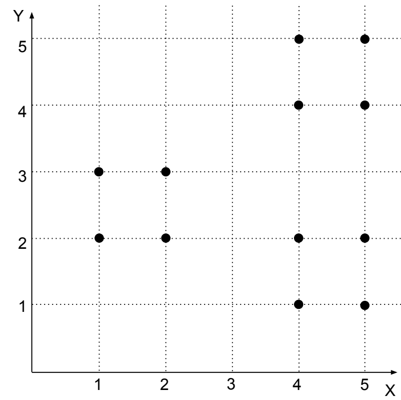

Figure 2: Reproduction of previous diagram.

(i) [5 points] Suppose we initialize k-means with the following cluster centroids:

$$\mu_1 = (5, 2) \qquad \mu_2 = (5, 4) \qquad \mu_3 = (5, 5)$$

and run k-means until convergence. What will the final cluster centroids be? You might find it easier to go through the k-means algorithm visually rather than grinding it out numerically.

Solution Initially, we assign points $\{(5,4),(4,4)\}$ to μ_2, $\{(5,5),(4,5)\}$ to μ_3, and the rest to μ_1. This gives us new centroids $(3,2),(4.5,4),(4.5,5)$, which cause convergence.

(ii) [5 points] What is the minimum reconstruction loss for $K = 3$ clusters on this data? Recall that for k-means, the reconstruction loss is defined as the sum over squared distances between points and centroids.

Solution Place a cluster centroid at the center of each of the three squares of points. This will give each point a squared distance of $1/2$. The total reconstruction loss $12 \cdot \frac{1}{2} = \boxed{6}$.

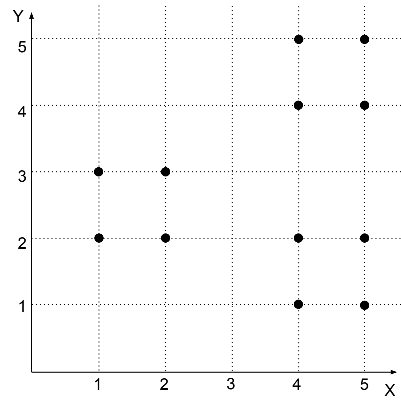

Figure 3: Reproduction of previous diagram.

(iii) [5 points] Give an initialization of three cluster centroids that yields the optimal reconstruction loss where the initialization coordinates are all **odd integers**. How many iterations of k-means will it take to converge to the optimal reconstruction loss?

Solution One possibility is: $\mu_1 = (1,3), \mu_2 = (5,1), \mu_3 = (5,5)$. With this initialization, k-means will converge in one iteration.

c. (*20 points*) **[Optimization and Losses]**

Alice and Bob are taking CS221 and trying to do some machine learning for their final project. They have a simple dataset consisting of the following $n = 20$ points (hopefully your data is a bit more interesting than this).

x:	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
y:	+	+	+	+	+	+	-	-	-	-	+	+	+	-	-	-	-	-	-	-

Bob learned about the hinge loss in lecture and thinks this is a good loss to minimize. Alice is concerned because they actually care about getting high accuracy, which is equivalent to getting low zero-one loss.

Define the linear predictor (parametrized by numbers w, b) to be

$$f(x) = \text{sign}(wx + b)$$

with the associated zero-one loss and hinge loss, respectively:

$$\text{Loss}_{\text{0-1}}(x, y, w, b) = \mathbf{1}[f(x) \neq y],$$
$$\text{Loss}_{\text{hinge}}(x, y, w, b) = \max(0, 1 - y(wx + b)).$$

Define the total training zero-one and hinge losses as the following:

$$\text{TrainLoss}_{\text{0-1}}(w, b) = \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \text{Loss}_{\text{0-1}}(x, y, w, b),$$
$$\text{TrainLoss}_{\text{hinge}}(w, b) = \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \text{Loss}_{\text{hinge}}(x, y, w, b).$$

(i) [4 points] What is the gradient[1] of the total hinge loss? More specifically, find the derivative of $\text{TrainLoss}_{\text{hinge}}$ with respect to w and with respect to b.

Solution The derivative of the max is an indicator function and then we use the chain rule and calculate the derivative of the argument.

$$\frac{d\text{TrainLoss}_{\text{hinge}}(w, b)}{dw} = \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \mathbf{1}[1 - y(wx + b) > 0](-yx)$$

$$\frac{d\text{TrainLoss}_{\text{hinge}}(w, b)}{db} = \sum_{(x,y)\in\mathcal{D}_{\text{train}}} \mathbf{1}[1 - y(wx + b) > 0](-y)$$

[1] The hinge loss is actually not differentiable at one point, but let's ignore this.

x:	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10
y:	+	+	+	+	+	+	-	-	-	-	+	+	+	-	-	-	-	-	-	-

(ii) [7 points] Suppose $w \in \{-1, +1\}$. Further suppose that b is **not an integer** so that the prediction is well-defined on all training points. Give a (w, b) pair that attains the global minimum of $\text{TrainLoss}_{\text{hinge}}$.

Solution We can try both values of w and set the gradient of b to 0 to find the optimum.

$$w = -1$$

$$b \in (-2, -1)$$

(iii) [3 points] What is the training zero-one loss of the parameters (w, b) you found in (ii)?

Solution With the above parameters, we are not able to correctly classify the following values of x : $\{-3, -2, 1, 2, 3\}$. This gives us a loss of 5.

(iv) [3 points] What is the minimum training zero-one loss over all possible (w, b), and give a value of (w, b) that attains this?

Solution Using $w = -1$, if we make $b \in (-4, -3)$, then we only misclassify $\{1, 2, 3\}$, which gives loss of 3. This is the minimum.

(v) [3 points] Alice is right in wanting to use the zero-one loss instead of the hinge loss because, in the end, we usually care about accuracy (which is a linear function of the zero-one loss). Give a reason why Bob is right in wanting to use the hinge loss instead of the zero-one loss.

Solution Two reasons: (1) unlike the zero-one loss, it sometimes has a non-zero gradient and (2) the hinge loss is convex

2. Bumpy car ride (*50 points*)

You're programming a self-driving car that can take you from home (position 1) to school (position n). At each time step, the car has a current position $x \in \{1, \ldots, n\}$ and a current velocity $v \in \{0, \ldots, m\}$. The car starts with $v = 0$, and at each time step, the car can either increase the velocity by 1, decrease it by 1, or keep it the same; this new velocity is used to advance x to the new position. The velocity is not allowed to exceed the speed limit m nor return to 0.

In addition, to prevent people from recklessly cruising down Serra Mall, the university has installed speed bumps at a subset of the n locations. The speed bumps are located at $B \subseteq \{1, \ldots, n\}$. The car is not allowed to enter, leave, or pass over a speed bump with velocity more than $k \in \{1, \ldots, m\}$. **Your goal is to arrive at position n with velocity 1 in the smallest number of time steps.**

Figure 4 shows an example with $n = 9$ positions and one speed bump $B = \{5\}$. If the maximum speed is $m = 3$ and $k = 1$ for a speed bump, then an example of a legal path is the following:

$$(1, 0) \xrightarrow{+1} (2, 1) \xrightarrow{+1} (4, 2) \xrightarrow{-1} (5, 1) \xrightarrow{0} (6, 1) \xrightarrow{+1} (8, 2) \xrightarrow{-1} (9, 1)$$

$x = 1$	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$	$x = 7$	$x = 8$	$x = 9$
home				bump				school

Figure 4: An example of a legal path that takes 6 time steps with $m = 3$ and $k = 1$. We show the position-velocity pairs (x, v) at each time step, and each number above an arrow is an acceleration (change in velocity).

a. (*10 points*)

(i) [6 points] Write a search problem whose minimum cost path corresponds to finding the fastest way to get from home to school (without violating speed limits). You should only use B, k, and m in your answer for Actions((x, v)) only, so this is where you should specify constraints that depend on B, k, and m. Be as precise as possible.

- $s_{\text{start}} = (1, 0)$
- $\text{Actions}((x, v)) =$
- $\text{Succ}((x, v), a) =$
- $\text{Cost}((x, v), a) =$
- $\text{IsEnd}((x, v)) =$

Solution

- $s_{\text{start}} = (1, 0)$
- $\text{Actions}((x, v)) = \{a \in \{+1, 0, -1\} : x + v + a \leq n \wedge v + a \leq m \wedge (v + a \leq k \vee \{x, \ldots, x + v + a\} \cap B = \emptyset)\}$. Suppose we want to apply acceleration a. First, we want to make sure we don't exceed the school ($x + v + a \leq n$) or go out of the velocity range ($v + a \leq m$). Next, we want to make sure that we're not entering, passing through, or leaving any speed bumps at a velocity greater than k. This is captured logically by ensuring a safe speed ($v + a \leq k$) or checking that there are no speed bumps between x and the new location $x + v + a$.
- $\text{Succ}((x, v), a) = (x + v + a, v + a)$
- $\text{Cost}((x, v), a) = 1$
- $\text{IsEnd}((x, v)) = \mathbf{1}[x = n \wedge v = 1]$

Most common errors were 1) not checking that there was a bump anywhere in between the starting position and the position after the action, and 2) not checking that the car does not go out of bounds after the action.

(ii) [4 points] Circle all the algorithms that are guaranteed to find the minimum cost path for this problem:

DFS BFS uniform cost search dynamic programming

Solution Since the cost of all the actions is 1 (a constant), we can use **BFS**. We can use **uniform cost search** since the action costs (1) is non-negative. Finally, we can also use **dynamic programming** because we are always moving in the forward direction, so the state graph is acyclic.

b. (*10 points*)

You want to use A* to compute the minimum cost path faster. Let's try to define a consistent heuristic based on solving a relaxed problem.

(i) [4 points] First, in the original problem P above, we required that the velocity of the car at one time step is within 1 of the velocity at the next time step. Define a relaxed search problem P_1 where we (i) allow the velocity of the car to change arbitrarily (but stay within the maximum velocities k, m), and (ii) we allow the car to arrive at n at any velocity instead of 1. In the example from Figure 4, we could take the following path, which would take only 4 time steps.

$$(1,0) \to (4,3) \to (5,1) \to (6,1) \to (9,3)$$

Define the state of the relaxed search problem P_1 as just the position x, and define $f_1(x)$ as the future cost from state x. Then from class, we know that $h_1((x,v)) = f_1(x)$ is a consistent heuristic for the original problem. Write the **recurrence** for $f_1(x)$:

Solution Formally

$$f_1(x) = \begin{cases} 0 & \text{if } x = n \\ 1 + \min_{v \in S_x} f_1(x+v) & \text{otherwise,} \end{cases}$$

where S_x contains the velocities $v \in \{1, \ldots, m\}$ satisfying $x + v \leq n$ and the speed limits: satisfying $v \leq k$ or $\{x, \ldots, x+v\} \cap B = \emptyset$.

(ii) [4 points] Define a further relaxed problem P_2 where we don't have to obey the speed limit regulations based on speed bumps at all (but still obey the overall speed limit m). Let $f_2(x)$ be the future cost under P_2, which gives rise to another consistent heuristic $h_2((x,v)) = f_2(x)$. Write a *closed form* expression for $f_2(x)$:

Solution Without any constraints, the fastest way is to go at maximum speed m to the goal, but where the last time step in general doesn't require velocity m.

$$f_2(x) = \left\lceil \frac{n-x}{m} \right\rceil.$$

(iii) [2 points] Discuss (in two or three sentences) the accuracy/speed tradeoffs of heuristics h_1 and h_2.

Solution Since P_2 is a relaxation of P_1, the heuristic h_1 (based on the future cost of P_1) is tighter than the heuristic h_2 (based on the future cost of P_2); formally $h_1((x,v)) \geq h_2((x,v))$, and thus it is more accurate. However, f_1 is more expensive to compute (requiring solving a recurrence), whereas f_2 can be computed in closed form.

c. (*10 points*)

It turns out that you were so excited about the AI algorithms that you didn't really pay much attention to the brakes of the car. As a result, when you try to decrease the velocity by 1, with some failure probability α, the velocity actually stays the same. To simplify our lives, assume there are no speed bumps. Assume a reward of R if we get to school (at a velocity of 1) but 0 if we pass the school, with a cost of 1 per time step. Let us formulate the resulting problem as an MDP:

- $s_{\text{start}} = (1, 0)$
- $\text{Actions}((x, v)) =$ same as that of (a)
- $T((x, v), a, (x', v')) =$ (to be filled out by you below)
- $\text{Reward}((x, v), a, (x', v')) = R \cdot \mathbf{1}[x' = n \wedge v' = 1] - 1$
- $\text{IsEnd}((x, v)) = \mathbf{1}[x \geq n]$

(i) [4 points] Fill out the definition of the transition probabilities T:

$T((x, v), a, (x', v')) =$

Solution

$$T((x, v), a, (x', v')) = \begin{cases} \alpha & \text{if } x' = x + v' \text{ and } v' = v \text{ and } a = -1 \\ 1 - \alpha & \text{if } x' = x + v' \text{ and } v' = v + a \text{ and } a = -1 \\ 1 & \text{if } x' = x + v' \text{ and } v' = v + a \text{ and } a \neq -1 \\ 0 & \text{otherwise.} \end{cases}$$

(ii) [6 points] Let us explore the effect of unreliable brakes. Consider the example in Figure 5.

$x = 1$ home	$x = 2$	$x = 3$	$x = 4$	$x = 5$ school

Figure 5: An small driving environment without speed bumps.

Consider two policies:

- π_1: always move with velocity 1:

$$\pi_1((1,0)) = +1 \quad \pi_1((2,1)) = 0 \quad \pi_1((3,1)) = 0 \quad \pi_1((4,1)) = 0.$$

- π_2: speed up and slow down:

$$\pi_1((1,0)) = +1 \quad \pi_1((2,1)) = +1 \quad \pi_1((4,2)) = -1.$$

Compute the expected utility of π_1 as a function of α and R (with discount $\gamma = 1$).

Solution The policy π_1 deterministically obtains reward $\boxed{R - 4}$. Using $Reward(x', v')$ we have $Reward(2,1) + Reward(3,1) + Reward(4,1) + Reward(5,1) = -1 - 1 - 1 - 1 + R$

Compute the expected utility of π_2 as a function of α and R (with discount $\gamma = 1$).

Solution The policy π_2 obtains reward $\boxed{(1-\alpha)R - 3}$. We only get reward R if we are able to break at the end so: $(1-\alpha)(Reward(2,1) + Reward(4,2) + Reward(5,1)) + \alpha$ $(Reward(2,1) + Reward(4,2) + Reward(6,2)) = (1-\alpha)(R-3) + \alpha(-3) = (1-\alpha)R - 3$

For what values of α and R does π_2 obtain higher expected reward than π_1? Your answer should be an expression relating α and R.

Solution Therefore, π_2 is better when $(1-\alpha)R - 3 > R - 4$, which is precisely when $\boxed{\alpha < 1/R}$.

d. (*10 points*)

Bad news: you realize that your brakes are not only faulty, but that you don't know how often they fail (α is unknown).

(i) [5 points] Circle all of the following algorithms that can be used to compute the optimal policy in this setting:

model-based Monte Carlo   model-free Monte Carlo   SARSA   Q-learning

Solution **Model-based Monte Carlo** would estimate the transition probabilities, which can be used to compute the optimal policy. **Q-learning** can be used to directly estimate the value of the optimal policy. Model-free Monte Carlo and SARSA can only be used to compute the value of a fixed policy.

(ii) [5 points] Suppose you want to estimate α and you drive around a bit and get the following episodes:

$$(1,0) \xrightarrow{+1} (2,1) \xrightarrow{0} (3,1) \xrightarrow{0} (4,1) \xrightarrow{0} (5,1) \xrightarrow{0} (6,1) \xrightarrow{0} (7,1)$$
$$(1,0) \xrightarrow{+1} (2,1) \xrightarrow{+1} (4,2) \xrightarrow{-1} (5,1) \xrightarrow{0} (6,1) \xrightarrow{0} (7,1)$$
$$(1,0) \xrightarrow{+1} (2,1) \xrightarrow{+1} (4,2) \xrightarrow{-1} (6,2) \xrightarrow{-1} (7,1)$$
$$(1,0) \xrightarrow{+1} (2,1) \xrightarrow{+1} (4,2) \xrightarrow{-1} (6,2) \xrightarrow{-1} (8,2)$$

What is the maximum likelihood estimate of α?

Solution First, note that α is "activated" whenever the action is $a = -1$. Success means that the velocity was decreased and failure means the velocity remains constant. Counting shows that there are 2 successes and 3 failures, so the maximum likelihood estimate of α is $\frac{3}{3+2} = \boxed{\frac{3}{5}}$. Note: this can be viewed as performing model-based Monte Carlo, but with parameter sharing (à la Bayesian network estimation).

e. (*10 points*)

You've fixed your brakes after missing class so many times (so $\alpha = 0$ now), but the university has caught on. **Let us return back to the conditions set forth in (a) and add some information.** The university wants to remove the old speed bumps and install a single new speed bump at location $b \in \{1, \ldots, n\}$ to maximize the time it takes for the car to go from position 1 to n.

Let $T(\pi, B)$ be the time it takes to get from 1 to n if the car follows policy π if speed bumps B are present. If π violates the speed limit, define $T(\pi, B) = \infty$.

To simplify, assume $n = 6$ and $k = 1$. Again, there is exactly one speed bump. That is, $B = \{b\}$ with $b \in \{1, \ldots, n\}$.

$x = 1$ home	$x = 2$	$x = 3$	$x = 4$	$x = 5$	$x = 6$ school

Figure 6: The university will add a speed bump somewhere.

(i) [5 points] Compute the worst case driving time, assuming you get to adapt your policy to the university's choice of speed bump location b: $\max_b \min_\pi T(\pi, \{b\})$. What values of b attain the maximum?

Solution Note that with $n = 6$, there are only two places where one can travel at a velocity of 2, from 2 to 4 or 3 to 5; in these cases, there can't be any speed bumps there. So if the speed bump is placed at $b \in \{1, 2, 5, 6\}$, the optimal policy has space to speed up to a velocity of 2 around the bump, so the total time is 4. However, if the speed bump is placed at $\boxed{b \in \{3, 4\}}$, then the optimal policy is to travel at a velocity of 1 the whole way which results in a total time of $\boxed{5}$, which is the worst case. Most common error was missing one of the cases for b. Also, there were a number of off-by-one errors (takes only 5 units to get from 1 to 6, not 6).

(ii) [5 points] Compute the best possible time assuming that you have to choose your policy before the university chooses the speed bump: $\min_\pi \max_b T(\pi, \{b\})$.

Solution If we choose any policy that has velocity of 2, the university can place the speed bump in the appropriate place that results in a time of ∞. Therefore, we must choose a policy that only has velocity 1, which results in a time of $\boxed{5}$. Many people took the min over the b location from the previous problem, but this is very distinct from the problem. This answer gives $\min_b \min_\pi$, while the problem asks for $\min_\pi \max_b$.

3. The Bayesian Bag of Candies Model (*50 points*)

You have a lot of candy left over from Halloween, and you decide to give them away to your friends. You have four types of candy: **A**pple, **B**anana, **C**aramel, **D**ark-Chocolate. You decide to prepare candy bags using the following process.

- For each candy bag, you first flip a (biased) coin Y which comes up heads ($Y = \mathrm{H}$) with probability λ and tails ($Y = \mathrm{T}$) with probability $1 - \lambda$.
- If Y comes up heads ($Y = \mathrm{H}$), you make a **H**ealthy bag, where you:
    1. Add one **A**pple candy with probability p_1 or nothing with probability $1 - p_1$;
    2. Add one **B**anana candy with probability p_1 or nothing with probability $1 - p_1$;
    3. Add one **C**aramel candy with probability $1 - p_1$ or nothing with probability p_1;
    4. Add one **D**ark-Chocolate candy with probability $1-p_1$ or nothing with probability p_1.
- If Y comes up tails ($Y = \mathrm{T}$), you make a **T**asty bag, where you:
    1. Add one **A**pple candy with probability p_2 or nothing with probability $1 - p_2$;
    2. Add one **B**anana candy with probability p_2 or nothing with probability $1 - p_2$;
    3. Add one **C**aramel candy with probability $1 - p_2$ or nothing with probability p_2;
    4. Add one **D**ark-Chocolate candy with probability $1-p_2$ or nothing with probability p_2.

For example, if $p_1 = 1$ and $p_2 = 0$, you would deterministically generate: **H**ealthy bags with one **A**pple and one **B**anana; and **T**asty bags with one **C**aramel and one **D**ark-Chocolate. For general values of p_1 and p_2, bags can contain anywhere between 0 and 4 pieces of candy.

Denote A, B, C, D random variables indicating whether or not the bag contains candy of type **A**pple, **B**anana, **C**aramel, and **D**ark-Chocolate, respectively.

a. (*15 points*)

(i) Draw the Bayesian network corresponding to process of creating a single bag.

Solution

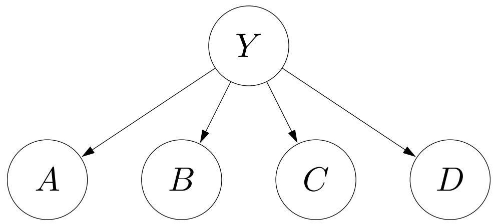

Figure 7: Bayesian network for a single candy bag.

(ii) What is the probability of generating a **H**ealthy bag containing **A**pple, **B**anana, **C**aramel, and not **D**ark-Chocolate? For compactness, we will use the following notation to denote this possible outcome:

$$(\textbf{H}\text{ealthy}, \{\textbf{A}\text{pple}, \textbf{B}\text{anana}, \textbf{C}\text{aramel}\}).$$

Solution By definition, we create a **H**ealthy bag with probability λ, and include the candies with probability $p_1 p_1 (1 - p_1) p_1$, so the result is

$$\lambda p_1 p_1 (1 - p_1) p_1$$

(iii) What is the probability of generating a bag containing **A**pple, **B**anana, **C**aramel, and *not* **D**ark-Chocolate?

Solution The bag could be **H**ealthy or **T**asty. We have computed the probability for the **H**ealthy case above. For a **T**asty one, a similar computation gives

$$(1 - \lambda) p_2 p_2 (1 - p_2) p_2$$

so the result is:

$$\lambda p_1 p_1 (1 - p_1) p_1 + (1 - \lambda) p_2 p_2 (1 - p_2) p_2$$

(iv) What is the probability that a bag was a **T**asty one, given that it contains **A**pple, **B**anana, **C**aramel, and *not* **D**ark-Chocolate?

Solution Using the definition of conditional probability, we get:

$$\frac{(1 - \lambda) p_2 p_2 (1 - p_2) p_2}{\lambda p_1 p_1 (1 - p_1) p_1 + (1 - \lambda) p_2 p_2 (1 - p_2) p_2}$$

b. (*10 points*)

You realize you need to make more candy bags, but you've forgotten the probabilities you used to generate them. So you try to estimate them looking at the 5 bags you've already made:

$$\begin{aligned}
&bag\ 1: && (\textbf{H}\text{ealthy}, \{\textbf{A}\text{pple}, \textbf{B}\text{anana}\})\\
&bag\ 2: && (\textbf{T}\text{asty}, \{\textbf{C}\text{aramel}, \textbf{D}\text{ark-Chocolate}\})\\
&bag\ 3: && (\textbf{H}\text{ealthy}, \{\textbf{A}\text{pple}, \textbf{B}\text{anana}\})\\
&bag\ 4: && (\textbf{T}\text{asty}, \{\textbf{C}\text{aramel}, \textbf{D}\text{ark-Chocolate}\})\\
&bag\ 5: && (\textbf{H}\text{ealthy}, \{\textbf{A}\text{pple}, \textbf{B}\text{anana}\})
\end{aligned}$$

Estimate λ, p_1, p_2 by maximum likelihood.

Solution Out of 5 bags, 3 are **H**ealthy, so $\lambda = 3/5$. To estimate p_1, we only consider the 3 healthy bags. For a **H**ealthy bag, the probability of adding **A**pple,**B**anana, not **C**aramel, and not **D**ark-Chocolateis $(p_1)^4$. For the three bags, the probability becomes $(p_1)^{12}$, which is maximized for $p_1 = 1$. Equivalently, to generate 3 **H**ealthy bags, we flip a (biased) coin of parameter p_1 12 times. Since we observe 12 "heads", the maximum likelihood estimate is $p_1 = 1$. To generate 2 **T**asty bags, we flip a (biased) coin of parameter p_2 8 times. Since we observe 0 "heads", the maximum likelihood estimate is $p_2 = 0$.

- $$\lambda = 3/5$$
- $$p_1 = 12/12 = 1$$
- $$p_2 = 0/8 = 0$$

Estimate λ, p_1, p_2 by maximum likelihood, using Laplace smoothing with parameter 1.

Solution We just need to increment the counts in the previous solution by 1.

- $$\lambda = 4/7$$
- $$p_1 = 13/(13 + 1)$$
- $$p_2 = 1/(1 + 9)$$

c. (*15 points*) You find out your little brother had been playing with your candy bags, and had mixed them up (in a uniformly random way). Now you don't even know which ones were **H**ealthy and which ones were **T**asty. So you need to re-estimate λ, p_1, p_2, but now without knowing whether the bags were **H**ealthy or **T**asty.

$$\begin{aligned}
&bag\ 1: && (?\ ,\ \{\textbf{A}\text{pple}, \textbf{B}\text{anana}, \textbf{C}\text{aramel}\}) \\
&bag\ 2: && (?\ ,\ \{\textbf{C}\text{aramel}, \textbf{D}\text{ark-Chocolate}\}) \\
&bag\ 3: && (?\ ,\ \{\textbf{A}\text{pple}, \textbf{B}\text{anana}, \textbf{C}\text{aramel}\}) \\
&bag\ 4: && (?\ ,\ \{\textbf{C}\text{aramel}, \textbf{D}\text{ark-Chocolate}\}) \\
&bag\ 5: && (?\ ,\ \{\textbf{A}\text{pple}, \textbf{B}\text{anana}, \textbf{C}\text{aramel}\})
\end{aligned}$$

You remember the EM algorithm is just what you need. Initialize with $\lambda = 0.5, p_1 = 0.5, p_2 = 0$, and run one step of the EM algorithm.

(i) E-step:

Solution To evaluate $P(Y = T \mid \{A, B, C\})$ we plug in the parameter values in the formula in (a),(iv), obtaining $P(Y = T \mid \{A, B, C\}) = 0$. To evaluate $P(Y = T \mid \{C, D\})$ we use a similar formula obtaining

$$P(Y = T \mid \{C, D\}) = \frac{(1-\lambda)(1-p_2)^4}{\lambda(1-p_1)^4 + (1-\lambda)(1-p_2)^4} = \frac{16}{17}$$

The resulting weighted dataset is:

- (**H**ealthy, $\{A, B, C\}$), 1×3
- (**T**asty, $\{A, B, C\}$), 0
- (**H**ealthy, $\{C, D\}$), $1/17 \times 2$
- (**T**asty, $\{C, D\}$), $16/17 \times 2$

(ii) M-step:

Solution Now we just do counts like in part (b). There are $3 + 2/17$ **H**ealthy bags out of 5. For p_1, each (**H**ealthy, $\{A, B, C\}$) corresponds to 3 "heads" and 1 "tail" (probability $p_1 p_1 (1-p_1) p_1$). Each (**H**ealthy, $\{C, D\}$) corresponds to 4 "tails" ($(1-p_1)^4$). For p_2, each (**T**asty, $\{C, D\}$) corresponds to 4 "tails" ($(1-p_2)^4$). The new parameters are:

$$\lambda = (3 + 2/17)/5$$

$$p_1 = 9/(9 + 3 + 4 * 2/17)$$

$$p_2 = 0$$

d. (*10 points*)

You decide to make candy bags according to a new process. You create the first one as described above. Then with probability μ, you create a second bag of the same type as the first one (Healthy or Tasty), and of different type with probability $1 - \mu$. Given this type, the bag is filled with candy as before. Then with probability μ, you create a third bag of the same type as the second one (Healthy or Tasty), and of different type with probability $1 - \mu$. And so on, you repeat the process M times. Denote Y_i, A_i, B_i, C_i, D_i the variables at each time step, for $i = 0, \dots, M$. Let $X_i = (A_i, B_i, C_i, D_i)$.

Now you want to compute:

$$\mathbb{P}(Y_i = \textbf{H}\text{ealthy} \mid X_0 = (1,1,1,0), \dots, X_i = (1,1,1,0))$$

exactly for all $i = 0, \dots, M$, and you decide to use the forward-backward algorithm.

Suppose you have already computed the marginals:

$$f_i = \mathbb{P}(Y_i = \textbf{H}\text{ealthy} \mid X_0 = (1,1,1,0), \dots, X_i = (1,1,1,0))$$

for some $i \geq 0$. Recall the first step of the algorithm is to compute an intermediate result *proportional* to

$$\mathbb{P}(Y_{i+1} \mid X_0 = (1,1,1,0), \dots, X_i = (1,1,1,0), X_{i+1} = (1,1,1,0))$$

(i) Write an expression that is **proportional** to

$$\mathbb{P}(Y_{i+1} = \textbf{H}\text{ealthy} \mid X_0 = (1,1,1,0), \dots, X_i = (1,1,1,0), X_{i+1} = (1,1,1,0))$$

in terms of f_i and the parameters p_1, p_2, λ, μ.

Solution

Emission: When $Y_{i+1} = \textbf{H}\text{ealthy}$, the probability of observing $X_{i+1} = (1,1,1,0)$ is $p_1 p_1 (1 - p_1) p_1$ as in part (a),(ii).

Transition: There are two cases: either $Y_i = \textbf{H}\text{ealthy}$, in which case we transit to $Y_{i+1} = \textbf{H}\text{ealthy}$ with probability μ, or $Y_i = \textbf{T}\text{asty}$, in which case we transit to $Y_{i+1} = \textbf{H}\text{ealthy}$ with probability $1 - \mu$.

$$\propto ((1 - f_i)(1 - \mu) + f_i \mu) p_1 p_1 (1 - p_1) p_1$$

(ii) Write an expression that is **proportional** to

$$\mathbb{P}(Y_{i+1} = \textbf{T}\text{asty} \mid X_0 = (1,1,1,0), \ldots, X_i = (1,1,1,0), X_{i+1} = (1,1,1,0))$$

in terms of f_i and the parameters of the model p_1, p_2, λ, μ. The proportionality constant should be the same as in (i).

Solution
(Similar to the previous question)
Emission: When $Y_{i+1} = \textbf{T}\text{asty}$, the probability of observing $X_{i+1} = (1,1,1,0)$ is $p_2 p_2(1 - p_2)p_2$.
Transition: There are two cases: either $Y_i = \textbf{H}\text{ealthy}$, in which case we transit to $Y_{i+1} = \textbf{T}\text{asty}$ with probability $1 - \mu$, or $Y_i = \textbf{T}\text{asty}$, in which case we transit to $Y_{i+1} = \textbf{T}\text{asty}$ with probability μ.

$$\propto ((f_i)(1-\mu) + (1-f_i)\mu)p_2p_2(1-p_2)p_2$$

(iii) Let h be the answer for part (i), and t for part (ii). Write an expression for

$$\mathbb{P}(Y_{i+1} = \textbf{H}\text{ealthy} \mid X_0 = (1,1,1,0), \ldots, X_i = (1,1,1,0), X_{i+1} = (1,1,1,0))$$

in terms of h, t and the parameters of the model p_1, p_2, λ, μ.

Solution Since h and t have same proportionality constant, we get the true value of the probability by normalization:

$$h/(h+t)$$

Congratulations, you have reached the end of the exam!