CS221 Exam Solutions

CS221
November 27, 2018

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	10	
	b	10	
	c	10	
	d	10	
	e	10	
2	a	10	
	b	10	
	c	10	
	d	10	
	e	10	
3	a	10	
	b	10	
	c	10	
	d	10	
	e	10	

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

1. **Wildlife** (*50 points*)

You are working in a wildlife conservation group, where you are installing n sensors to detect the presence of a wild animal called a pangolin. Let $Y \in \{0, 1\}$ denote whether there is actually a pangolin (in the given location), and let $X_1, \ldots, X_n$ denote the predicted outputs of the n sensors, where each $X_i \in \{0, 1\}$. We assume the following:

- There is a natural rate of pangolin appearance $p(y = 1) = h$.
- All sensors have the same false positive rate of $p(x_i = 1 \mid y = 0) = \alpha$.
- All sensors have the same false negative rate of $p(x_i = 0 \mid y = 1) = \beta$.
- All sensor outputs are conditionally independent given the actual appearance (see Figure **??**).

Figure 1: A Bayesian network with $n = 4$ sensors relating the presence of a pangolin Y and sensor outputs $X_1, \ldots, X_4$.

a. (*10 points*)

The specification sheet for the sensors does not provide the false positive rates or the false negative rates, so you have to estimate them. Each week, you install a new sensor, and record all the outputs of all the sensors installed thus far. Then you go out into the field and observe whether there is a pangolin or not (Y). Below is the data you have collected, where "-" indicates there is no data for that sensor for that week.

	X_1	X_2	X_3	X_4	Y
Week 1	0	-	-	-	0
Week 2	0	1	-	-	0
Week 3	1	1	0	-	1
Week 4	0	0	1	0	0

Compute the maximum likelihood estimate of the parameters (h, α, β) of the Bayesian network:

$h =$

$\alpha =$

$\beta =$

Solution The maximum likelihood solution has a closed form which can be computed by simply counting and normalizing:

$$h = \frac{\text{number of } Y = 1}{\text{total number of weeks}} = \frac{1}{4}, \tag{1}$$

$$\alpha = \frac{\text{number of } Y = 0 \wedge X_i = 1}{\text{number of } Y = 0} = \frac{2}{7}, \tag{2}$$

$$\beta = \frac{\text{number of } Y = 1 \wedge X_i = 0}{\text{number of } Y = 1} = \frac{1}{3}. \tag{3}$$

Note that each week gives rise to a variable number of counts, but this does not matter.

b. (*10 points*)

After experiencing the dangers of data collection in the wild for four weeks, you decide that it's not really your true calling. Instead, you're just going to use your estimated parameters (h, α, β) to predict Y for future weeks.

Suppose that you have n sensors installed. During one week, you observe that all the sensors output 1. What is $\mathbb{P}(Y = 1 \mid X_1 = \cdots = X_n = 1)$, i.e. the probability that there is a pangolin given the observations?

Your answer should be an expression defined in terms of (h, α, β, n).

Solution Let's compute a more general answer where $0 \leq k \leq n$ of the sensors output 1. Consider $Y = 1$. For each of the k X_i's that are observed to be 1, we have $p(x_i = 1 \mid y = 1) = 1 - \beta$. For the rest of the $n - k$ X_i's that are observed to be 0, we have $p(x_i = 0 \mid y = 1) = \beta$. We can do an analogous calculation for $Y = 0$ with α instead of β. The result is as follows:

$$\mathbb{P}(Y = 1 \mid X_1 = x_1, \ldots, X_n = x_n) \tag{4}$$

$$= \frac{p(y = 1) \prod_{i=1}^{n} p(x_i \mid y = 1)}{p(y = 1) \prod_{i=1}^{n} p(y = 1 \mid x_i) + p(y = 0) \prod_{i=1}^{n} p(x_i \mid y = 0)} \tag{5}$$

$$= \frac{h(1 - \beta)^k \beta^{n-k}}{h(1 - \beta)^k \beta^{n-k} + (1 - h)\alpha^k (1 - \alpha)^{n-k}} \tag{6}$$

$$= \boxed{\frac{h(1 - \beta)^n}{h(1 - \beta)^n + (1 - h)\alpha^n}}, \tag{7}$$

where in the last line, we used the fact that $k = n$.

c. (*10 points*)

Your boss asks you to install more sensors, but you first want to understand what happens as the number of sensors goes to infinity. Note: your solution will depend strongly on your formula for part (b).

(i) [5 points] Suppose $\alpha + \beta < 1$. What does $\mathbb{P}(Y = 1 \mid X_1 = \cdots = X_n = 1)$ converge to as $n \to \infty$? You should explicitly state how your answer depends on h (your answer should cover every possible value of $0 \leq h \leq 1$). You should also give a rigorous mathematical justification for your answer.

Solution Let us take the solution for part (b) and divide both the numerator and denominator by $h(1-\beta)^n$:

$$\mathbb{P}(Y = 1 \mid X_1 = x_1, \ldots, X_n = x_n) \tag{8}$$

$$= \frac{1}{1 + \frac{(1-h)\alpha^n}{h(1-\beta)^n}}. \tag{9}$$

We can now more clearly see that as long as $h > 0$ and as long as $\alpha < 1 - \beta$, this quantity will converge to $\boxed{1}$ as $n \to \infty$. Intuitively, as you get more evidence of sensors outputting 1, your confidence in $Y = 1$ should increase. If $h = 0$, then the probability is $\boxed{0}$ for all n. Intuitively, if you start closed-minded, no evidence will change your mind.

(ii) [5 points] Suppose $\alpha + \beta = 1$. What does $\mathbb{P}(Y = 1 \mid X_1 = \cdots = X_n = 1)$ converge to as $n \to \infty$? You should explicitly state how your answer depends on h (your answer should cover every possible value of $0 \leq h \leq 1$). You should also give a rigorous mathematical justification for your answer.

Solution If $\alpha + \beta = 1$, then $(1-\beta)^n = \alpha^n$, so therefore the expression from part (b) simplifies to:

$$\mathbb{P}(Y = 1 \mid X_1 = x_1, \ldots, X_n = x_n) = \frac{h}{h + (1-h)} = \boxed{h}. \tag{10}$$

In this case, your sensors are intuitively worthless and provide no information about Y, and we are simply left with the prior probability h.

d. (*10 points*)

Let us try to solve the classification problem (predicting Y from $X_1, \ldots, X_n$) in a different way now. We could use the standard hinge loss or logistic loss, but we want to design our loss function to capture the fact that false negatives are much more unacceptable than false positives (we don't want to miss any pangolins!).

Let us suppose we define a linear classifier as follows:[1]

$$f(x) = [\mathbf{w} \cdot \phi(x) \geq 0]. \tag{11}$$

Define a loss function as follows:

$$\text{Loss}(x, y, \mathbf{w}) = \begin{cases} \ell_1(x, y, \mathbf{w}) & \text{if } y = 1 \\ \ell_0(x, y, \mathbf{w}) & \text{if } y = 0. \end{cases} \tag{12}$$

Design ℓ_0 and ℓ_1 based on the hinge loss so that the following properties hold:

- If the margin of a point (x, y) (defined as $\mathbf{w} \cdot \phi(x)$ for $y = 1$ and $-\mathbf{w} \cdot \phi(x)$ for $y = 0$) is at least 1, then the loss is zero on that point.
- If two points $(x_1, 1)$ and $(x_2, 0)$ have the same margin, then the loss of $(x_1, 1)$ is 5 times that of $(x_2, 0)$.

(i) [5 points] Specify the loss function:

$\ell_0(x, y, \mathbf{w}) =$

$\ell_1(x, y, \mathbf{w}) =$

Solution We define the hinge loss on $y = 0$ and 5 times the hinge loss on $y = 1$:

$$\ell_1(x, y, \mathbf{w}) = 5 \max(0, 1 - \mathbf{w} \cdot \phi(x)), \tag{13}$$

$$\ell_0(x, y, \mathbf{w}) = \max(0, 1 + \mathbf{w} \cdot \phi(x)). \tag{14}$$

Note that you can't multiply the score by y to get the margin anymore because $y \in \{0, 1\}$, not $y \in \{-1, 1\}$.

[1] Note that the output label is $y \in \{0, 1\}$ rather than $y \in \{1, -1\}$ as we've seen in class.

(ii) [5 points] As usual, we need to compute the gradient to optimize this objective. Compute the gradient of the loss function with respect to $\mathbf{w}$. Express your answer in terms of $\mathbf{w}, \phi(x), y$.

$\nabla \text{Loss}(x, y, \mathbf{w}) =$

Solution The gradient can be computed separately for each case:

$$\nabla \text{Loss}(x, y, \mathbf{w}) = \begin{cases} -5\phi(x) & \text{if } y = 1 \wedge \mathbf{w} \cdot \phi(x) < 1 \\ \phi(x) & \text{if } y = 0 \wedge \mathbf{w} \cdot \phi(x) > -1 \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

We won't be too picky about whether the inequalities are strict or not.

e. (*10 points*)

Suppose that the feature vector is $\phi(x) = [1, \sum_{i=1}^{n} x_i]$, where $x_i \in \{0, 1\}$ is still the output of sensor i. Consider the 4 examples from part (a), where an unobserved value is treated as a zero:

	X_1	X_2	X_3	X_4	Y
Week 1	0	-	-	-	0
Week 2	0	1	-	-	0
Week 3	1	1	0	-	1
Week 4	0	0	1	0	0

For example, for week 3, $\phi(x) = [1, 2]$.

For this two-dimensional feature vector, the weight vector is $\mathbf{w} = (w_1, w_2)$. If we set $w_1 = -5$, what are all the possible values of w_2 for which the resulting weight vector $\mathbf{w}$ achieves zero training loss according to the loss function you defined in part (d)?

Solution Zero training loss is equivalent to each example achieving zero loss. Obtaining zero loss on an example is equivalent to imposing a inequality constraint on the weight vector $\mathbf{w}$. Each of the 4 data points imposes a constraint that $\mathbf{w} \cdot \phi(x) \geq 1$ for $y = 1$ and $\mathbf{w} \cdot \phi(x) \leq -1$ for $y = 0$:

$$(w_1 + w_2 \cdot 0) \leq -1 \tag{16}$$
$$(w_1 + w_2 \cdot 1) \leq -1 \tag{17}$$
$$(w_1 + w_2 \cdot 2) \geq 1 \tag{18}$$
$$(w_1 + w_2 \cdot 1) \leq -1 \tag{19}$$

Simplifying, we have the following (the second and fourth constraints are the same):

$$w_1 \leq -1 \qquad w_1 + w_2 \leq -1 \qquad w_1 + 2w_2 \geq 1 \tag{20}$$

So if $w_1 = -5$, then this means $w_2 \leq 4$ and $w_2 \geq 2$. So we conclude that $\boxed{3 \leq w_2 \leq 4}$.

2. **Driving** (*50 points*)

Optimal Driving Corporation (ODC) has hired you as a consultant to help design optimal algorithms for their new fleet of self-driving cars. They are currently focused on a particular stretch of freeway with H lanes and length W (see Figure **??**). There are $K+1$ cars on the road: Your car (the one being controlled algorithmically) starts at position $x_0^{(0)} = (1, 1)$, and the K other cars start at positions $x_1^{(0)}, \ldots, x_K^{(0)}$.

Each car (yours and others) has the same set of four possible actions (the successor states below are for the car in $(5, 2)$ from Figure **??**):

- $+(1, 0)$: Move forward one step (ending up in $(6, 2)$)
- $+(2, 0)$: Move forward two steps (ending up in $(7, 2)$)
- $+(1, -1)$: Move forward one step and to the left (ending up in $(6, 1)$)
- $+(1, 1)$: Move forward one step and to the right (ending up in $(6, 3)$)

A car cannot take an action that takes it out of one of the H lanes; for example, a car in $(1, 3)$ cannot take action $+(1, 1)$.

If a car leaves this stretch of freeway (e.g., taking action $+(1, 0)$ from $(7, 2)$), then the car is teleported to a special end position $\odot$. Once in $\odot$, any action a car takes will keep it in $\odot$ and incur 0 cost.

All cars take turns moving, starting with your car, followed by cars 1 through K, and then your car, etc. The game ends when all cars have left the stretch of freeway (moved into position $\odot$).

Each action (except for those taken from position $\odot$) has a base cost of 1. There are also a set of potholes P. If at the end of an action, a car lands on a pothole or in a position occupied by another car, it incurs an additional cost of c (and does not affect any other cars). Each car's goal is to minimize its own total cost. For example, if your car takes actions that lead to the following state sequence: $(1, 1) \to (2, 2) \to (4, 2) \to (6, 2) \to (7, 3)$, the cost incurred by you is $1 + 1 + (1 + c) + 1 = 4 + c$. Note that cost is only incurred based on where a car lands, so the cost of $(2, 2) \to (4, 2)$ above is 1, even though there is a pothole at $(3, 2)$.

Figure 2: An example of the driving scenario, where we have a freeway of length $W = 7$ and $H = 3$ lanes. Your car starts at $x_0^{(0)} = (1, 1)$ and the other $K = 3$ cars start in $x_1^{(0)} = (1, 3)$, $x_2^{(0)} = (3, 1)$, and $x_3^{(0)} = (5, 2)$. The potholes $P = \{(2, 1), (3, 2), (3, 3), (4, 1), (6, 2)\}$ are shown as gray ellipses. The arrows show one possible action that each of the cars can take.

a. (*10 points*)

Let us formulate this as a $(K+1)$-player game. Define the state to be $(x_0, \dots, x_K, j)$, where x_0 is the position of your car, $x_1, \dots, x_K$ are the positions of other cars, and $j \in \{0, \dots, K\}$ specifies which car's turn it is.

Rather than defining only a utility for end states for each car as in class, we define a cost for all state-action pairs: in particular, let $\text{Cost}_j(s, a)$ be the cost incurred by player j when action a is taken from state s. Complete the game specification below.

- $s_{\text{start}} = (x_0^{(0)}, \dots, x_K^{(0)}, 0)$
- $\text{Player}((x_0, \dots, x_K, j)) = j$
- $\text{Actions}((x_0, \dots, x_K, j)) = \{a \in \{+(1,0), +(2,0), +(1,-1), +(1,1)\} : \text{InLane}(x_j + a)\}$, where $\text{InLane}((u,v)) = [(u,v) = \odot \text{ or } 1 \le v \le H]$. By convention, we define $\odot + a = \odot$, so that cars that have left the stretch of freeway stay in the same end state.
- $\text{Succ}((x_0, \dots, x_K, j), a) =$

Solution

$$\text{Succ}((x_0, \dots, x_K, j), a) = (x_0, \dots, x_{j-1}, x_j + a, x_{j+1}, \dots, x_K) \tag{21}$$

- $\text{Cost}_j((x_0, \dots, x_K, j), a) =$

Solution

$$\text{Cost}_j((x_0, \dots, x_K, j), a) = \tag{22}$$
$$1 + c \cdot [x_j + a \in P \vee x_j + a = x_{j'} \text{ for some } j' = 0, \dots, K] \tag{23}$$

- $\text{IsEnd}(x_0, \dots, x_K, j) =$

Solution

$$\text{IsEnd}(x_0, \dots, x_K, j) = [x_{j'} = \odot \text{ for all } j' = 0, \dots, K] \tag{24}$$

b. (*10 points*)

Assume the other K cars follow the deterministic policy of always moving forward one step (choosing $a = (1, 0)$).

(i) [5 points] Compute the maximum possible number of unique states $(x_0, \dots, x_K, j)$ which are reachable from a given s_{start}, where your car can take any action and the other K cars take only the action given by the deterministic policy. Your answer should be as tight an (asymptotic) upper bound as possible. You must provide the expression that has the correct (smallest) dependence on W, H, K, but don't worry about constants (if the true answer is $W - 1$, then $O(W)$, $2W$, W are all acceptable, but W^2 is not).

Solution There are $K + 1$ cars, each of which could be in WH positions, which would result in $(WH)^{K+1}$, but this is an overestimate. The key is to note that the other K cars move deterministically, so their positions are purely a function of how many actions they have taken. Therefore, we just need to keep track of the following:

- The position of your car x_0 ($WH + 1$ possible values)
- Whose turn it is j ($K + 1$ possible values)
- The minimum number of actions that any car has taken (at most $W - 1$ possible values, since each action moves a car forward by at least 1).

So the total number of reachable states is at most $\boxed{(WH + 1)(K + 1)(W - 1) = O(W^2HK)}$.

(ii) [5 points] What if your car was no longer allowed to move forward two steps (i.e., $(2, 0) \notin \text{Actions}((x_0, \dots, x_K, 0))$)? Compute the (asymptotic) maximum possible number of unique states that are reachable in this case.

Solution In this case, all the cars move forward at the same rate, so x_0 (in particular, the horizontal coordinate of x_0) and the turn j determines the position of the other cars. Therefore, the number of states is $\boxed{WHK}$. Note that this is equivalent to saying that we could have reduced our state space to (x_0, j).

c. (*10 points*)

It is Sunday morning and there are no other cars on the road ($K = 0$). In this case, the problem is simply a search problem over the state x_0. You realize that you might be able to compute the optimal policy faster by leveraging A*. But of course to do that, we need a consistent heuristic.

(i) [5 points] Design a non-trivial consistent heuristic that can be computed in closed form, and prove why your heuristic is consistent. Here, your car position $x_0 = (u, v)$.

$h((u, v)) =$

Solution Recall that we can obtain a consistent heuristic by defining a relaxed problem. We can assume that there are no potholes ($P = \emptyset$), which does not increase the cost of any action. Now, the coast is clear and we simply need to compute the number of moves to get from (u, v) to $\odot$, which is a Manhattan distance of $W - u + 1$. Each action reduces the Manhattan distance by 1 or 2, so the minimum number of actions is

$$\boxed{h(u, v) = \left\lceil \frac{W - u + 1}{2} \right\rceil}. \tag{25}$$

(ii) [5 points] Prove that the following heuristic is inconsistent:

$$h((u, v)) = \sum_{u'=u+1}^{W} c \cdot \mathbb{1}[(u', v) \in P], \tag{26}$$

which simply counts the number of potholes between $(u + 1, v)$ and (W, v) and multiplies by the cost c of landing on a pothole (recall that $\mathbb{1}$ is the indicator function). Hint: try to construct a simple example.

Solution Recall that all consistent heuristics have to be admissible, which means that $h(x_0) \leq \text{FutureCost}(x_0)$. However, to show inconsistency, we just have to construct an example where the heuristic is not an underestimate of the minimum cost. Let $W = 2$ and $H = 1$, $P = \{(2, 1)\}$, $c = 2$, and $x_0 = (1, 1)$. Then the minimum cost is $\text{FutureCost}(x_0) = 1$, but the heuristic is $h(x_0) = c$, which is larger.

d. (*10 points*)

It turns out that your assumption that all other cars just move forward by one step ($a = +(1, 0)$) is wrong (shockingly). You still assume that each car has the same policy, but this policy chooses an action only depending on what's at the four positions a car could move into. Suppose we have a car at position x_j and it's j's turn. Define a feature vector $\phi((x_0, \ldots, x_K, j))$, which has two features for each action a:

1. whether there is a pothole at $x_j + a$, and
2. whether there is another car at $x_j + a$.

Figure 3: Repeat of Figure **??** for convenience. The training data here is the action taken (shown by the arrow) for each of the three other cars.

For example, for the car at $x_j = (3, 1)$ in Figure **??**, which takes action $+(2, 0)$, the feature vector would be the following:

name	value
$+(1,0)$ has pothole:	1
$+(1,0)$ has car:	0
$+(2,0)$ has pothole:	0
$+(2,0)$ has car:	0
$+(1,-1)$ has pothole:	0
$+(1,-1)$ has car:	0
$+(1,1)$ has pothole:	0
$+(1,1)$ has car:	0

As shorthand, write this feature vector as 10000000.

Note that for the purposes of computing features, positions off the road are treated as empty positions.

(i) [5 points] We will first use a "tabular" method for estimating the policy. The idea is that for each value of the feature vector, we will estimate a distribution over possible actions based on the data, which is the actions of the three other cars in Figure **??**, given by the arrows. In other words, our training data are three (feature vector, action) pairs. Specify only the non-zero probabilities and use the shorthand notation for the features. We have filled in the feature vector and action for the car at $x_j = (3, 1)$ for you.

Hint: think about how we perform model-based estimation of transition probabilities for MDPs, given observations of (state, action, successor) triples.

feature vector ϕ	action a	estimated probability $\pi(a \mid (x_0, \dots, x_K, j))$
10000000	(+2, 0)	

Solution

feature vector ϕ	action a	estimated probability $\pi(a \mid (x_0, \dots, x_K, j))$
10000000	$+(1, 1)$	$\frac{1}{2}$
10000000	$+(2, 0)$	$\frac{1}{2}$
00100000	$+(1, 0)$	1

(ii) [5 points] Next you consider using a linear predictor with the same features ϕ. In one or two sentences, state one advantage and one disadvantage of doing so over using the tabular approach above.

Solution The linear predictor has fewer parameters (2(number of actions) $= 8$) as opposed to $2^{2(\text{number of actions})} = 256$. Fewer parameters means that there's less of a chance of overfitting, but more parameters means that with enough data, we learn a more accurate model. In this particular example, the number of parameters of the tabular approach isn't that big, so we shouldn't need that much data. On the other hand, a linear model is perhaps a reasonable approximation of a car's behavior, which should be sufficient to capture basic rough strategies like avoiding potholes and other cars and trying to move forward as much as possible.

e. (*10 points*)

In practice, we don't know what policies the other cars could be facing. For each setting, specify what is the most specific type of model that could be used to represent it (e.g., search problem, MDP, two-player zero-sum game, CSP, etc.). For example, if something is a search problem, don't say MDP, even though all search problems are special cases of MDPs. What inference algorithm would you use for the model (e.g., uniform cost search, minimax, etc.)? If we have not covered inference algorithms for a particular model, say so. Simply state the model and inference algorithm, and briefly justify your answer.

1. (2 points) Each other car's policy chooses the action a that minimizes its immediate cost (not your cost) plus the distance from $x_i + a$ to (W, H). Any ties are broken randomly.

   Solution The greedy policy is a known stochastic policy. Due to the randomness, it is not a search problem, but can be cast as an MDP. The default algorithm for computing optimal policies is value iteration. But because the MDP is acyclic, we could also compute this using a recursive dynamic programming.

2. (2 points) Each other car's policy uses the learned policy produced from part (d).

   Solution This is again a known stochastic policy (the fact that it was learned is irrelevant), so the answer is the same as for the greedy policy.

3. (2 points) Each other car's policy is optimally minimizing your cost (which might be the case if you had a siren on your car).

   Solution All cars are now trying to minimize your cost, so this is a search problem. Which can be solved using UCS or A* (all costs are non-negative), or dynamic programming (since the state graph is acyclic).

4. (2 points) Each other car's policy is optimally maximizing your cost.

   Solution This is a classic turn-based zero-sum game (very similar to Pac-Man with cars instead of ghosts). We would compute the recurrence using dynamic programming.

5. (2 points) Each other car's policy is trying to minimize its own cost.

   Solution This is a turn-based non-zero-sum game. These games in general don't have optimal policies, but merely Nash equilibria. How to compute them is outside the scope of this class.

3. **Farm** (*50 points*)

Farmer Kim wants to install a set of sprinklers to water all his crops in the most cost-effective manner and has hired you as a consultant. Specifically, he has a rectangular plot of land, which is broken into $W \times H$ cells. For each cell (i, j), let $C_{i,j} \in \{0, 1\}$ denote whether there are crops in that cell that need watering. In each cell (i, j), he can either install $(X_{i,j} = 1)$ or not install $(X_{i,j} = 0)$ a sprinkler. Each sprinkler has a range of R, which means that any cell within Manhattan distance of R gets watered. The maintenance cost of the sprinklers is the sum of the Manhattan distances from each sprinkler to his home located at $(1, 1)$. Recall that the Manhattan distance between (a_1, b_1) and (a_2, b_2) is $|a_1 - a_2| + |b_1 - b_2|$. Naturally, Farmer Kim wants the maintenance cost to be as small as possible given that all crops are watered. See Figure **??** for an example.

Figure 4: An example of a farm with $W = 5$ and $H = 3$. Each cell (i, j) is marked 'C' if there are crops there that need watering $(C_{i,j} = 1)$. An example of a sprinkler installation is given: a cell (i, j) is marked with 'S' if we are placing a sprinkler there $(X_{i,j} = 1)$. Here, the sprinkler range is $R = 1$, and the cells that are shaded are the ones covered by some sprinkler. In this case, the sprinkler installation is valid (all crops are watered), and the total maintenance cost is $1 + 4 = 5$.

a. (*10 points*)

Farmer Kim actually took CS221 years ago, and remembered a few things. He says: "I think this should be formulated as a factor graph. The variables should be $X_{i,j} \in \{0, 1\}$ for each cell (i, j). But here's where my memory gets foggy. What should the factors be?"

Let $X = \{X_{i,j}\}$ denote a full assignment to all variables $X_{i,j}$. Your job is to define two types of factors:

- $f_{i,j}$: ensures any crops in (i, j) are watered,
- f_{cost}: encodes the maintenance cost,

so that a maximum weight assignment corresponds to a valid sprinkler installation with minimum maintenance cost.

$f_{i,j}(X) =$

Solution For each cell (i, j), let $f_{i,j}$ encode whether the crops (if they exist) in (i, j) are watered:

$$f_{i,j}(X) = \left[C_{i,j} = 0 \text{ or } \min_{i',j' : X_{i',j'}=1} |i' - i| + |j' - j| \leq R \right]. \tag{27}$$

$f_{\text{cost}}(X) =$

Solution We define the next factor to the exponentiated negative minimum cost, so that the factor is non-negative and that maximizing the weight corresponds to minimizing the maintenance cost:

$$f_{\text{cost}}(X) = \exp\left(- \sum_{i',j' : X_{i',j'}=1} |i' - 1| + |j' - 1| \right). \tag{28}$$

b. (*10 points*)

Every year, Farmer Kim plants a different set of crops in possibly different cells; in other words $\{C_{i,j}\}$ changes year by year. Farmer Kim says: "Maybe I can just start with my assignment $X = \{X_{i,j}\}$ from last year and run Gibbs sampling to try to improve it to a better solution."

(i) [5 points] Recall that Gibbs sampling sets $X_{i,j} = 1$ with some probability p. For convenience, use the notation $X \cup \{X_{i,j} : 1\}$ to denote a modification of X where $X_{i,j}$ has been assigned 1 (analogously for 0). Write an expression for p in terms of the factors (e.g., f_{cost}). Your expression should involve as *few* factors as possible.

$p =$

Solution The only factors that depend on $X_{i,j}$ are f_{cost} and the $f_{i',j'}$ of any cells (i', j') within range of R. Let's multiply only those factors together for a candidate choice $v \in \{0, 1\}$:

$$w_v = f_{\text{cost}}(X \cup \{X_{i,j} : v\}) \prod_{i',j':|i'-i|+|j'-j'|\leq R} f_{i',j'}(X \cup \{X_{i,j} : v\}). \tag{29}$$

Then the probability p is just proportional to that:

$$p = \frac{w_1}{w_1 + w_0}. \tag{30}$$

(ii) [5 points] Recall Gibbs sampling is guaranteed to find the optimal assignment eventually if there is a non-zero probability of reaching any valid assignment X' from the initial assignment X. Prove that this is the case for any X, X'.

Solution Note that Gibbs sampling chooses an assignment with probability proportional to its weight, so therefore it cannot choose an assignment with weight 0. We must show that we can reach any assignment without going through any zero-weight assignment. The key insight is adding sprinklers to an assignment X with non-zero weight cannot make its weight zero (although it can decrease its weight). Take any two assignments X, X' with non-zero weight. We can construct a path through intermediate assignments with non-zero weight as follows: add sprinklers one by one until all of them are added, and then remove sprinklers one by one until we obtain X'. Note that this is only one such positive probability path, which is probably not the best one, but it suffices to prove the claim.

c. (*10 points*)

Having installed the sprinklers ($X_{i,j}$), Farmer Kim wants to install water sources on K cells, $\mu_1, \ldots, \mu_K$, to power these sprinklers. Each sprinkler ((i,j) for which $X_{i,j} = 1$) is to be assigned to a particular water source $z_{i,j} \in \{1, \ldots, K\}$. The *transportation cost* of an installation is the sum of the Manhattan distances from each sprinkler ((i,j) for which $X_{i,j} = 1$) to its assigned water source.

For example, in Figure **??**, if $K = 1$, we might install one water source $\mu_1 = (3, 2)$, which would obtain a transportation cost of $2 + 1 = 3$.

Similar to K-means, derive an alternating minimization algorithm that alternates between minimizing the transportation cost with respect to water source assignments (step 1) and minimizing with respect to the location of the water sources (step 2).

(i) [5 points] Step 1: given $\mu_1, \ldots, \mu_K$, write an expression for $z_{i,j}$ that minimizes the transportation cost. Notation: let $\mu_k[0]$ and $\mu_k[1]$ be the two coordinates of μ_k.

Solution This update is the same as in k-means:

$$z_{i,j} = \arg\min_{k=1,\ldots,K} |\mu_k[0] - i| + |\mu_k[1] - j|. \tag{31}$$

Note that this needs to be done for cells (i,j) for which $X_{i,j} = 1$.

(ii) [5 points] Step 2: given the sprinkler to water source assignments $\{z_{i,j}\}$, write an expression for μ_k that minimizes the transportation cost. What is the time complexity for calculating ALL μ_k? Give your answer in big-O notation as a function of W, H, K.

Solution This update is the same as in k-means, except that μ_k is restricted to be one of the cells, not an arbitrary point. Therefore, this is a discrete optimization problem rather than a continuous one.

$$\mu_k = \arg\min_{1 \le i \le W, 1 \le j \le H} \sum_{(i',j') : X_{i,j} = 1, z_{i,j} = k} |i' - i| + |j' - j|. \tag{32}$$

This optimization problem can be solved by enumerating over the WH cells and computing the sum, and then taking the cell with the smallest sum. For each K, we can loop over all possible candidate means (i,j) (WH of them), and loop over the data points (at most WH of them). So the time complexity is $\boxed{O(KW^2H^2)}$. Note that you could be more efficient with more clever data structures.

d. (*10 points*)

A year has passed after Farmer Kim installed the sprinklers, and now some of the sprinklers have broken. Let $B_{i,j} \in \{0, 1\}$ denote whether a sprinkler at cell (i, j) is broken. Farmer Kim has called you back and is asking for help. He wants to run an irrigation pipe starting from $(1, 1)$ and running through a subset of the broken sprinklers and back to $(1, 1)$.

Every cell that has watered crops generates r dollars of revenue, but connecting a pipe from one cell to an adjacent cell costs c dollars. Farmer Kim naturally wants to maximize profit (revenue minus cost).

Figure 5: Repeat of Figure **??** for convenience, except now with both sprinklers broken.

For example, in Figure **??**, suppose both sprinklers are broken ($B_{2,1} = B_{4,2} = 1$). Here are two options:

1. Fix both sprinklers: build the pipe $(1, 1) \to (2, 1) \to (3, 1) \to (4, 1) \to (4, 2) \to (4, 1) \to (3, 1) \to (2, 1) \to (1, 1)$. This would cost $8c$ and produce all the crops for a revenue of $3r$, so the profit would be $3r - 8c$.
2. Fix the $(2, 1)$ sprinkler: build the pipe $(1, 1) \to (2, 1) \to (1, 1)$. This would cost $2c$ and produce crops at $(3, 1)$ and $(2, 2)$ for a revenue of $2r$, so the profit would be $2r - 2c$.

Your job is to define a search problem where the minimum cost path corresponds to the maximum profit. Farmer Kim also declares that you must fix at least one sprinkler. Farmer Kim has already told you that the state is $((i, j), S)$, where (i, j) is the current position and S is the set of sprinklers (cells) that are working so far. Complete the following search problem specification. *You should give an intuitive description as well as being mathematically precise using the given notation.*

- $s_{\text{start}} = ((1, 1), S_0)$, where $S_0 = \{(i, j) : X_{i,j} = 1 \text{ and } B_{i,j} = 0\}$.
- $\text{Actions}(((i, j), S)) = \{a \in \{(-1, 0), (+1, 0), (0, -1), (0, +1)\} : (i, j) + a \text{ is in bounds}\}$
- $\text{IsEnd}(((i, j), S)) = [i = 1 \text{ and } j = 1 \text{ and } S \neq S_0]$, where we finish when we reach $(1, 1)$ having fixed at least one sprinkler.
- $\text{Succ}(((i, j), S), a) =$

  Solution First, the new position is $(i, j) + a$. Second, we add the new position to the set S if it is a bad sprinkler.

  $$\text{Succ}(((i, j), S), a) = \begin{cases} ((i, j) + a, S) & \text{if } B_{(i,j)+a} = 0 \\ ((i, j) + a, S \cup \{(i, j) + a\}) & \text{if } B_{(i,j)+a} = 1 \end{cases} \tag{33}$$

- $\text{Cost}(((i, j), S), a) =$

  Solution First, we pay a fixed cost of c for extending the pipe by one cell. If a leads us to a new position with a bad sprinkler, then that sprinkler gets fixed, and we reap a reward r of any crop that has wasn't watered before, but is watered now. Thus,

  $$\text{Cost}(((i, j), S), a) = c - r \left[B_{(i,j)+a} = 1\right] \ |\Delta|, \tag{34}$$
  $$\Delta = \{(i', j') : C_{i',j'} = 1 \tag{35}$$
  $$\text{and } \neg\text{covers}(S, (i', j')) \tag{36}$$
  $$\text{and covers}(S \cup \{(i, j) + a\}, (i', j'))\}, \tag{37}$$
  $$\text{covers}(S, (i, j)) = \left[\min_{i',j' \in S} |i' - i| + |j' - j| \leq R\right]. \tag{38}$$

e. (*10 points*)

Suppose that the revenue per crop r is much larger than the cost c; specifically, assume $r > 2(W + H - 2)c$. Assume that sprinklers have been placed $X = \{X_{i,j}\}$ to minimize the maintenance cost as in part (a). In terms of the solution to part (d), will it always be optimal to fix all the broken sprinklers? Either prove the claim or show a counterexample.

Solution First, note that since $X = \{X_{i,j}\}$ is the optimum solution to part (d), each sprinkler is watering some crop that is not covered by any other sprinkler, or else we would be able to simply remove this sprinkler and satisfy all the constraints, violating the fact that X would not be optimal. Therefore, fixing a broken sprinkler generates at least r revenue. At the same time, the additional cost of fixing this sprinkler is at most $2(W+H-2)c$, which would be running a pipe from $(1,1)$ to the farthest corner (W,H) and back. By assumption, $r > 2(W + H - 2)c$, so the net profit generated by fixing this sprinkler is positive.

Congratulations, you have reached the end of the exam!