CS 188 Spring 2021 — Introduction to Artificial Intelligence

# Written HW 3

**Due:** Friday 03/19/2021 at 10:59pm (submit via Gradescope).

**Policy:** Can be solved in groups (acknowledge collaborators) but must be written up individually

**Submission:** It is recommended that your submission be a PDF that matches this template. You may also fill out this template digitally (e.g. using a tablet). **However, if you do not use this template, you will still need to write down the below four fields on the first page of your submission.**

| | |
|---|---|
| First name | |
| Last name | |
| SID | |
| Collaborators | |

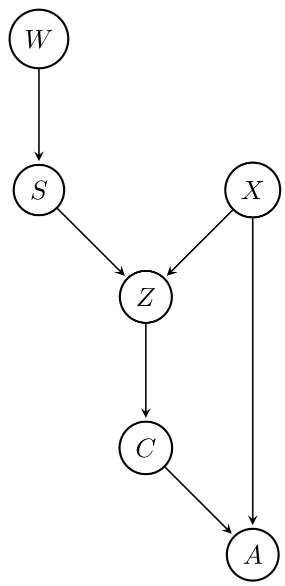
**For staff use only:**

| | | |
|---|---|---|
| Q1. | Quadcopter: Spectator | /30 |
| Q2. | Quadcopter: Data Analyst | /55 |
| Q3. | Localization with HMMs | /15 |
| | Total | /100 |

# Q1. [30 pts] Quadcopter: Spectator

Flying a quadcopter can be modeled using a Bayes Net with the following variables:

- $W$ (weather) $\in$ {clear, cloudy, rainy}
- $S$ (signal strength) $\in$ {strong, medium, weak}
- $X$ (true position) $= (x, y, z, \theta)$ where $x, y, z$ **each** can take on values $\in \{0, 1, 2, 3, 4\}$ and $\theta$ can take on values $\in \{0°, 90°, 180°, 270°\}$
- $Z$ (reading of the position) $= (x, y, z, \theta)$ where $x, y, z$ **each** can take on values $\in \{0, 1, 2, 3, 4\}$ and $\theta$ can take on values $\in \{0°, 90°, 180°, 270°\}$
- $C$ (control from the pilot) $\in$ {forward, backward, rotate left, rotate right, ascend, descend} (6 controls in total)
- $A$ (smart alarm to warn pilot if that control could cause a collision) $\in$ {bad, good}




**(a)** Representation

**(i)** [3 pts] What is $N_x$, where $N_x$ is the domain size of the variable $X$? Please explain your answer.

Answer: $N_x =$ ☐

Explanation: ☐

**(ii)** [4 pts] Please list **all** of the Conditional Probability Tables that are needed in order to represent the Bayes Net above. Note that there are 6 of them.

**(iii)** [3 pts] What is the size of the Conditional Probability Table for $Z$? You may use $N_x$ in your answer.

Now, assume that we look at this setup from the perspective of Spencer – a spectator who can observe A and W. Spencer observes $A$=bad and $W$=clear, and he now wants to infer the signal strength. In BN terminology, he wants to calculate $P(S|A = \text{bad}, W = \text{clear})$.

**(b)** [5 pts] Inference by Enumeration

If Spencer chooses to solve for this quantity using inference by enumeration, how many unique probability terms need to be multiplied together?

**(c)** [15 pts] Inference by Variable Elimination

Spencer chooses to solve for this quantity by performing variable elimination in the order of $Z - X - C$. Answer the following prompts to work your way through this procedure.

(1a) First, we need to eliminate $Z$. Which factors (from the 6 CPTs above) are involved?

(1b) Describe how you eliminate the variable of interest by multiplication of those factors. What conditional probability **factor** results from this step?

(2a) Second, we need to eliminate $X$. Which factors are involved?

(2b) Describe how you eliminate the variable of interest by multiplication of those factors. What conditional probability **factor** results from this step?

(3a) Third, we need to eliminate $C$. Which factor/s are involved?

(3b) Describe how you eliminate the variable of interest by multiplication of those factors. What conditional probability **factor** results from this step?

(4) List the 3 conditional probability factors that you calculated as a result of the 3 elimination steps above, along with their domain sizes. You may use $N_x$ in your answer. Which factor is the biggest? Is this bigger or smaller than the biggest factor from the "inference by enumeration" approach?

(5) List the **1** unused conditional probability factor from the 3 that you calculated above, and also list the **2** resulting conditional probability factors from the 6 original CPTs.

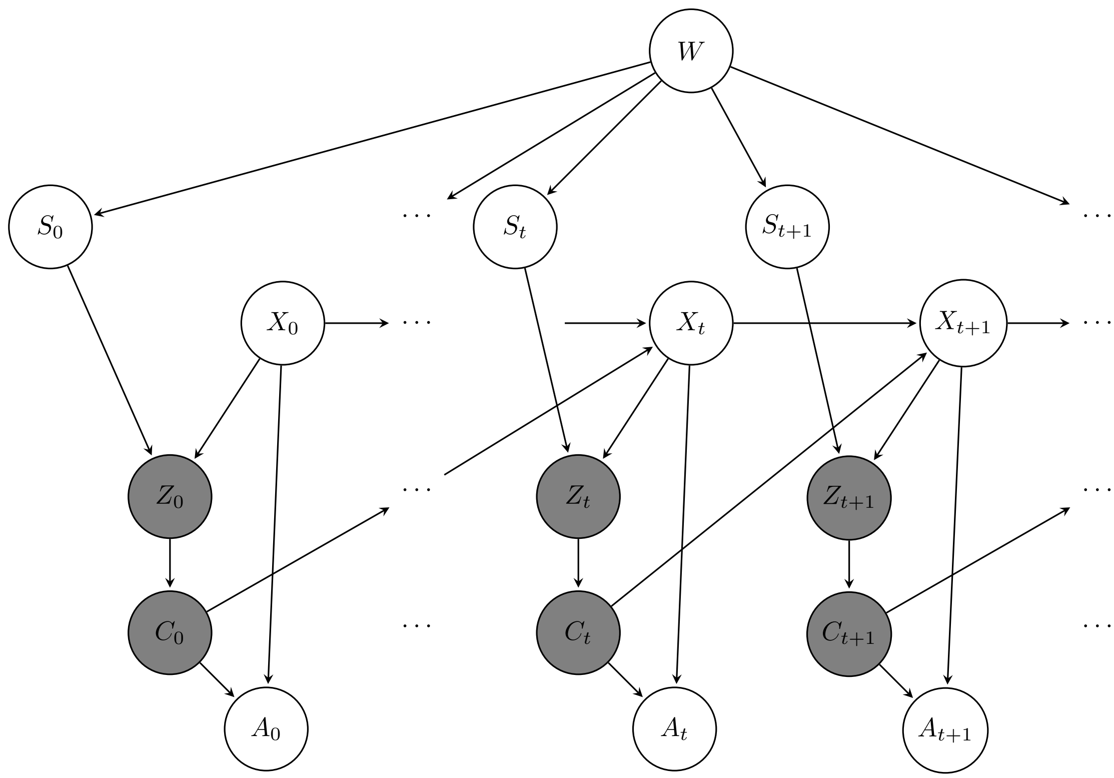
(6) Finally, let's solve for the original quantity of interest: $P(S|A = \text{bad}, W = \text{clear})$. After writing the equations to show how to use the factors from (5) in order to solve for $f(S|A = bad, W = clear)$, don't forget to write how to turn that into a probability $P(S|A = bad, W = clear)$.
Hint: use Bayes Rule, and use the 3 resulting factors that you listed in the previous question.

# Q2. [55 pts] Quadcopter: Data Analyst

In this question, we look at the setup from the previous problem, but we consider the quadcopter flight over time. Here, flight can be considered in discrete time-steps: $t \in 0, 1, 2, ..., N-1$ with, for example, $X_t$ representing the true position $X$ at discrete time-step $t$. Suppose the weather (W) does not change throughout the quadcopter flight.

One key thing to note here is that there are edges going between time $t$ and time $t+1$: The true position at time $t+1$ depends on the true position at time $t$ as well as the control input from time $t$.

Let's look at this setup from the perspective of Diana, a data analyst who can only **observe** the output from a data-logger, which stores **Z (reading of position) and C (control from the pilot)**.




**(a)** Hidden Markov Model

**(i)** [4 pts] List all the hidden variables and observed variables in this setup. In a few sentences, how is this setup different from the vanilla Hidden Markov Model you saw in lecture? You should identify at least 2 major differences.

Hidden variables:

Observed variables:

Differences:

**(ii)** [3 pts] As a data analyst, Diana's responsibility is to infer the true positions of the quadcopter throughout its flight. In other words, she wants to find a list of true positions $x_0, x_1, x_2, ..., x_{N-1}$ that are the most likely to have happened, given the recorded readings $z_0, z_1, z_2, ..., z_{N-1}$ and controls $c_0, c_1, c_2, ..., c_{N-1}$.

What is the objective expressed in **conditional probability**?

**(iii)** [3 pts] Can you expressed part (ii) in **joint probability**? Why or why not?

**(b)** Forward Algorithm Proxy

Conner, a colleague of Diana's, would like to use this model (with the $Z_t$ and $C_t$ observations) to perform something analogous to the forward algorithm for HMMs to infer the true positions. Let's analyze below the effects that certain decisions can have on the outcome of running the forward algorithm.

**(i)** [5 pts] He argues that since W (weather) does not depend on time, and is not something he is directly interested in, he does not need to include it in the forward algorithm, so he replaces $P(S_t|W)$ everywhere with $P(S_t)$, where $P(S_t) = \sum_W P(S_t|W)P(W)$. Is this correct? If not, what effect does it have?

**(ii)** [10 pts] He also argues that he does not need to include hidden state A (smart alarm warning) in the forward algorithm. What effect does not including A in the forward algorithm have on (a) the accuracy of the resulting belief state calculations, and on (b) the efficiency of calculations? Please justify your answer.

Accuracy:

Efficiency:

**(iii)** [5 pts] Last but not least, Conner recalls that for the forward algorithm, one should calculate the belief at time-step $t$ by conditioning on evidence up to $t-1$, instead of conditioning on evidence from the entire trajectory (up to $N-1$). Let's assume that some other algorithm allows us to use evidence from the full trajectory ($t = 0$ to $t = N-1$) in order to infer each belief state. What is an example of a situation (in this setup, with the quadcopter variables) that illustrates that incorporating evidence from the full trajectory can result in better belief states than incorporating evidence only from the prior steps?

- ◯ If the signal strength is bad before $t-1$, but gets better later.
- ◯ If the signal strength is good up to $t-1$, and the signal is lost later.
- ◯ There isn't such example because using evidence up to $t-1$ gives us the optimal belief.

**(c)** Forward Algorithm: The real deal

Now we look at this question from the perspective of Paul, a quadcopter pilot who can **observe W (weather), Z (reading of position), C (control from the pilot), and A (smart alarm warning)**.

**(i)** [5 pts] Now that the only hidden states are $S_t$ and $X_t$, is this graph a well-behaving HMM (where $E_{t+1} \perp\!\!\!\perp E_t \mid X_{t+1}$ and $X_{t+1} \perp\!\!\!\perp E_t \mid X_t$, recall that $X$ is the hidden variable and $E$ is the evidence variable)? Please explain your reasoning.

**(ii)** [10 pts] What is the **time-elapsed update** from time-step $t$ to time-step $t+1$? Be sure to include all hidden states and observed states, and show how to assemble the update from the conditional probability tables corresponding to the graph. Denote $f(S_t, X_t) = P(S_t, X_t \mid W, Z_{0:t}, C_{0:t}, A_{0:t})$. Find $f'(S_{t+1}, X_{t+1}) = P(S_{t+1}, X_{t+1} \mid W, Z_{0:t}, C_{0:t}, A_{0:t})$. Justify your choice.

- ○ $f'(S_{t+1}, X_{t+1}) = \max_{s_t} \sum_{x_t} P(S_{t+1} \mid W) * P(X_{t+1} \mid x_t, c_t) * f(S_t, X_t)$
- ○ $f'(S_{t+1}, X_{t+1}) = \sum_{s_t} \sum_{x_t} P(S_{t+1} \mid S_t) * P(X_{t+1} \mid x_t) * f(S_t, X_t)$
- ○ $f'(S_{t+1}, X_{t+1}) = \sum_{s_t} \sum_{x_t} P(Z_{t+1} | S_{t+1}, X_{t+1}) P(S_{t+1} \mid S_t) * P(X_{t+1} \mid x_t) * f(S_t, X_t)$
- ○ $f'(S_{t+1}, X_{t+1}) = \sum_{s_t} \sum_{x_t} P(S_{t+1} \mid S_t) * P(X_{t+1} \mid x_t) P(X_{t+1} \mid c_t) * f(S_t, X_t)$
- ○ $f'(S_{t+1}, X_{t+1}) = \sum_{s_t} \sum_{x_t} P(S_{t+1} \mid W) * P(X_{t+1} \mid x_t, c_t) * f(S_t, X_t)$
- ○ $f'(S_{t+1}, X_{t+1}) = \sum_{s_t} \max_{x_t} P(S_{t+1} \mid W) * P(X_{t+1} \mid x_t, c_t) * f(S_t, X_t)$

**(iii)** [10 pts] What is the **observation update** at time-step $t+1$? Be sure to include all hidden states and observed states, and show how to assemble the update from the conditional probability tables corresponding to the graph. Denote $f'(S_{t+1}, X_{t+1}) = P(S_{t+1}, X_{t+1} \mid W, Z_{0:t}, C_{0:t}, A_{0:t})$. Find $f(S_{t+1}, X_{t+1})$. Justify your choice.

- ○ $f(S_{t+1}, X_{t+1}) = P(Z_{t+1} \mid S_{t+1}, X_{t+1}) * P(C_{t+1} \mid Z_{t+1}) * P(A_{t+1} \mid C_{t+1}) * f'(S_{t+1}, X_{t+1})$
- ○ $f(S_{t+1}, X_{t+1}) = P(Z_{t+1} \mid S_{t+1}, X_{t+1}) * P(C_{t+1} \mid Z_{t+1}) * f'(S_{t+1}, X_{t+1})$
- ○ $f(S_{t+1}, X_{t+1}) = P(Z_{t+1} \mid S_{t+1}, X_{t+1}) * P(C_{t+1} \mid Z_{t+1}) * P(A_{t+1} \mid C_{t+1}, X_{t+1}) * f'(S_{t+1}, X_{t+1})$
- ○ $f(S_{t+1}, X_{t+1}) \propto P(Z_{t+1} \mid S_{t+1}, X_{t+1}) * P(C_{t+1} \mid Z_{t+1}) * P(A_{t+1} \mid C_{t+1}, X_{t+1}) * f'(S_{t+1}, X_{t+1})$
- ○ $f(S_{t+1}, X_{t+1}) \propto P(Z_{t+1} \mid S_{t+1}, X_{t+1}) * P(C_{t+1} \mid Z_{t+1}) * f'(S_{t+1}, X_{t+1})$

# Q3. [15 pts] Localization with HMMs

In this question, you will answer a few questions about HMM-based localization and fill in a few lines of code in the google colab posted on piazza to see localization in action.

**(a)** **(i)** [5 pts] Recall the filtering algorithm in HMMs:

$$P(X_{t+1}|e_{1:t+1}) = \alpha P(e_{t+1}|X_{t+1}) \sum_{x_t} P(x_t|e_{1:t}) P(X_{t+1}|x_t)$$

where $P(X_{t+1}|x_t)$ is the transition probability, $P(x_t|e_{1:t})$ is the current distribution over possible states at time $t$ given evidence $e_{1:t}$, and $P(e_{t+1}|X_{t+1})$ is the probability of observing evidence at the next timestep $e_{t+1}$ given our next state $X_{t+1}$. Describe how the filtering algorithm can be vectorized (done in parallel) using matrix-matrix and/or matrix-vector multiplications. Define your matrices/vectors, and write an equation for the filtering algorithm update rule in terms of those matrices and vectors.

**(ii)** [1 pt] Take a look at the docstring in `init_hmm_model(...)`. What are the transition probabilities for the robot at $(x, y)$ where there is a wall to the North of $(x, y)$?

**(iii)** [1 pt] Take a look at `robot_sensor.get_sensor_info(...)` and `get_error_vector(...)`. Briefly describe the sensor used by the robot and what kinds of sensor reading errors can occur.

**(b)** Coding Portion:

**(i)** [2 pts] Fill in `create_observation_matrix(...)` in the `HMM` class by defining the `probability` variable. Given the `error_rate` and a list of number of wall-reading errors made by the sensor `num_errors`, calculate the probability $P(e_{1:t+1}|X_{t+1})$ and store this value in the variable `probability`. (Use `num_errors[i]` to get the number of sensor errors made given that the robot is at state $i$.) Hint: The probability is modeled via a binomial distribution involving `error_rate` and `num_errors[i]`. You may find `comb(n, k)` helpful for calculating $\binom{n}{k}$. You will not need to do anything else in this function.

**(ii)** [3 pts] Fill in `filtering(...)` in the `HMM` class. You will use `observation_matrix`, `self.current_state`, and `self.transition_matrix` to run one step of filtering, as represented by your answer in part a(i). After performing filtering on your current state, put the new value in `self.current_state`. Note that `self.current_state` is a probability distribution over all non-wall spaces and should be normalized to sum to one to account for the $\alpha$ term in the filtering update. Hint: Our solution is 2 lines long and does not use any loops, since it is vectorized. You will not need to implement anything else in this function.

**(iii)** [0 pts] **Submission Directions**: On your colab, please go to `File` → `Download .py`, and submit this python file to gradescope. We will set up an autograder for the coding portion after the due date. Note that the autograder will not run if you do not submit a python file!

**(c)** **(i)** [1 pt] Observe the rollouts of localization for the 4x4 maze, 10x10 maze, and 25x25 maze. Note that red squares are walls, white squares are non-walls, and gray squares in the left subplots indicate the belief distribution (a visual coloring of `self.current_state` probabilities, where black = probability 1 that the robot is at that location, and white = probability 0). The green square in the right subplot indicates the robot ground truth location. Comment on how well the `self.current_state` probabilities model the ground truth robot location at timesteps $t = 1$ and $t = 20$ for the 10x10 maze.

**(ii)** [2 pts] Compare the localization task in this assignment with that of the logic project. What are some algorithmic differences and similarities between logical inference-based localization and probabilistic HMM-based localization?