CS 188 Spring 2024 | Introduction to Artificial Intelligence | HW 5B Sol.

Solutions for HW 5B

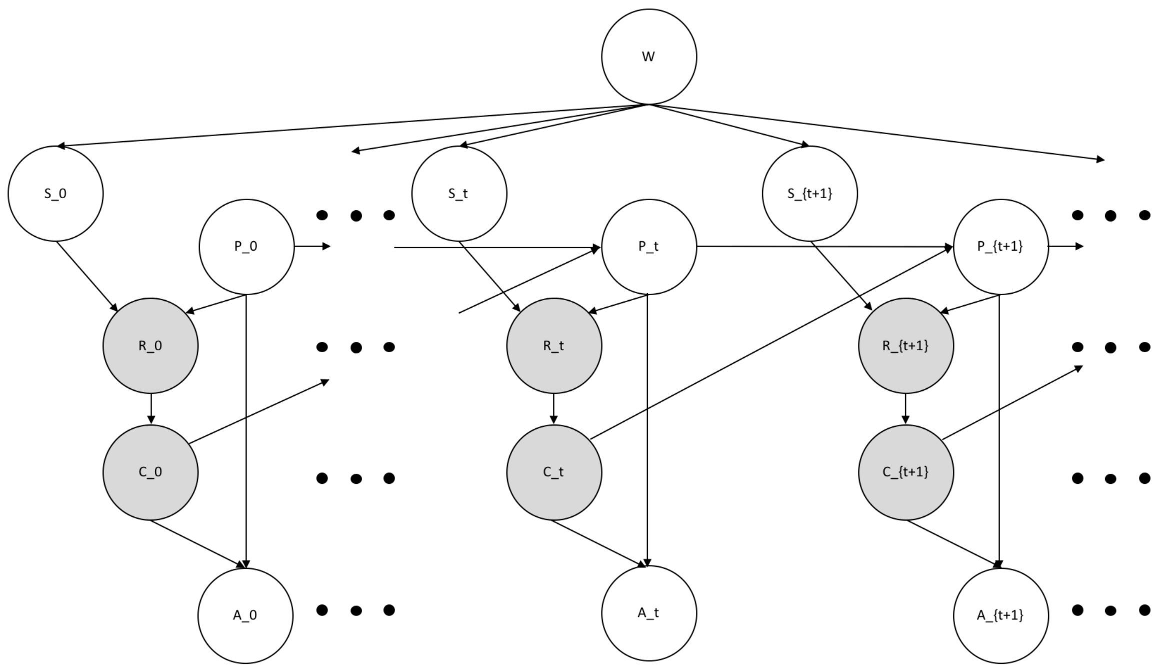
Q1. [31 pts] Quadcopter: Data Analyst

As in the last homework, we will consider a Bayes net for quadcopter flight. Our Bayes net has the following variables: W (weather), S (signal strength), P (true position), R (reading of the position), C (control from the pilot), and A (smart alarm to warn pilot if that control could cause a collision).

We will also 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, P_t representing the true position P 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 **R (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: W, $S_i \forall i$, $P_i \forall i$, $A_i \forall i$

Observed variables: $R_i \forall i$, $C_i \forall i$

Differences: Include but not limited to

1. There is one overarching hidden variable that doesn't change with respect to time.
2. There are multiple observation variables and multiple hidden variables at any time step.
3. There is a hidden variable (A) at the tail of observed variables (R and C).

(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 $p_0, p_1, p_2, ..., p_{N-1}$ that are the most likely to have happened, given the recorded readings $r_0, r_1, r_2, ..., r_{N-1}$ and controls $c_0, c_1, c_2, ..., c_{N-1}$.

Write down the probability that Diana tries to maximize in terms of a **joint probability**, and interpret the meaning of that probability. Note that the objective that you write below is such that Diana is solving the following problem: $\max_{p_0, p_1, \ldots, p_N}$ (maximization objective).

Maximization objective: $P(P_0 = p_0, P_1 = p_1, \ldots, P_{N-1} = p_{N-1}, R_0 = r_0 \ldots R_{N-1} = r_{N-1}, C_0 = c_0 \ldots C_{N-1} = c_{N-1})$

Explanation: Probability of the positions being there and the observations of reading/controls being there also.

(iii) [3 pts] Morris, a colleague of Diana's, points out that maximizing the joint probability is the same as maximizing a **conditional probability** where all evidence ($r_0, r_1, r_2, \ldots$ and $c_0, c_1, c_2, \ldots$) are moved to the right of the conditional bar. Is Morris right?

- ● Yes, and I will provide a proof/explanation below.
- ○ No, and I will provide a counter example below.

Yes. $max_p P(p, r, c)$ is the same as $max_p P(p|r, c)$ because when you factor the joint, the $P(r, c)$ is independent of the optimization variables p and can thus be considered a constant.

(b) The Markov Property

(i) [5 pts] In this setup, conditioned on all observed evidence, does the sequence $S_0, S_2, ..., S_{N-2}$ follow the Markov property? Please justify your answer. No because S_{t+1} is not conditionally independent of S_{t-1} given S_t. We can see this by just examining the subgraph containing S_{t-1}, W, and S_{t+1}. None of the Bayes net independence rules apply in this case.

(c) Forward Algorithm Proxy

Conner, a colleague of Diana's, would like to use this model (with the R_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.

Note that when we say to **not include** some variable in the algorithm, we mean that we marginalize/sum out that variable. For example, if we do not want to include W in the algorithm, then we replace $P(S_t|W)$ everywhere with $P(S_t)$, where $P(S_t) = \sum_W P(S_t|W)P(W)$.

(i) [4 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. What effect does not including W 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: This makes everything less informed than if you were to include the W, so accuracy is worse. Another way to explain by not including W, we make the assumption that S_t are independent of each other, which is an incorrect assumption.

Efficiency: Efficiency is better since you only need to include $P(S_t)$ everywhere, instead of $P(S_t|W)$. Another way to explain is that $P(S_t)$ has fewer entries than $P(S_t|W)$

(ii) [3 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: Since A is downstream of the observations and also doesn't effect the next timesteps in any way, not including A in the forward algorithm has no effect on accuracy.

Efficiency: No effect of efficiency, since it shouldn't be included anyway. Note: if student says that it reduces efficiency because of the extra marginalization step mentioned above (since you wouldn't have done that otherwise), consider that correct, even though you don't need to do it anyway.

(iii) [3 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.

If the signal strength is bad at the beginning of the traj but gets better later, the information from later in the trajectory can greatly improve the belief state estimates earlier in the trajectory (whereas if only evidence up to $t-1$ is allowed to be used at step t, then those early belief states would not have been good).

(d) Policy Reconstruction

Emily, another colleague of Diana's, would like to use this model to reconstruct the pilot's policy from data. Let's analyze below the effects that certain decisions can have on the outcome of doing policy reconstruction.

(i) [2 pts] Emily states that the probabilistic model for the pilot's **policy** is entirely captured in one Conditional Probability Table from the Bayes Net Representation. Which table do you think this is, and explain why this table captures the pilot's policy.

Table: $P(C|R)$

Explanation: The probability of control given a reading of position is exactly what a policy looks like in this representation

(ii) [4 pts] Emily argues that if we were given a lot of data from the data logger, we could reconstruct the probabilistic model for the pilot's policy. Is she right?

- ● Yes, and I will provide an overview of how to reconstruct the pilot's policy from the data.
- ○ No, and I will provide a list of reasons for why we cannot reconstruct the policy.

Counting is your friend. We can reconstruct the policy by filling in a table of R and C where each entry is the number of times (r, c) appear in the data. We then use this table to construct a CPT table for $P(C|R)$.