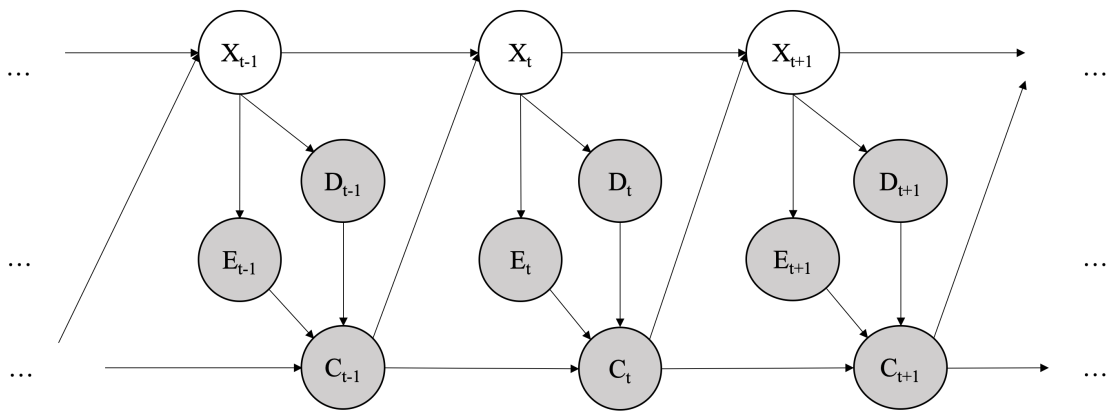
CS 188
Summer 2022

Exam Prep 5A Solutions

Q1. We Are Getting Close...

The CS 188 TAs have built an autonomous vehicle, and it's finally on the street! Approaching a crossroad, our vehicle must avoid bumping into pedestrians. However, how close are we?
X is the signal received from sensors on our vehicle. We have a estimation model E, which estimates the current distance of any object in our view. Our vehicle also needs a model to detect objects and label their classes as one of {pedestrian, stop sign, road, other}. The TAs trained a detection model D that does the above and with a simple classification, outputs one of {no pedestrian, pedestrian on the road, pedestrian beside the stop sign}. Our vehicle has a control operator C, which determines the velocity by changing the acceleration.



(a) For the above Dynamic Bayes Net, complete the equations for performing updates. (Hint: think about the prediction update and observation update equations in the forward algorithm for HMMs.)

Time elapse: **(i)** = **(ii)** **(iii)** **(iv)** $P\left(x_{t-1}|e_{0:t-1}, d_{0:t-1}, c_{0:t-1}\right)$

(i) ○ $P(x_t)$ ● $P\left(x_t|e_{0:t-1}, d_{0:t-1}, c_{0:t-1}\right)$ ○ $P\left(e_t, d_t, c_t|e_{0:t-1}, d_{0:t-1}, c_{0:t-1}\right)$

(ii) ○ $P(c_{0:t-1})$ ○ $P(x_{0:t-1}, c_{0:t-1})$ ○ $P(e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$
○ $P(e_{0:t}, d_{0:t}, c_{0:t})$ ● 1

(iii) ● $\Sigma_{x_{t-1}}$ ○ Σ_{x_t} ○ $\max_{x_{t-1}}$ ○ $\max_{x_t}$ ○ 1

(iv) ○ $P(x_{t-1}|x_{t-2})$ ○ $P(x_{t-1}, x_{t-2})$ ○ $P(x_t|e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$
○ $P(x_t|x_{t-1})$ ○ $P(x_t, x_{t-1})$ ○ $P(x_t, e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$
● $P(x_t|x_{t-1}, c_{t-1})$ ○ $P(x_t, x_{t-1}, c_{t-1})$ ○ 1

Recall the prediction update of forward algorithm: $P(x_t|o_{0:t-1}) = \Sigma_{x_{t-1}} P(x_t|x_{t-1}) P\left(x_{t-1}|o_{0:t-1}\right)$, where o is the observation. Here it is similar, despite that there are several observations at each time, which means o_t corresponds to e_t, d_t, c_t for each t, and that X is dependent on the C value of the previous time, so we need $P(x_t|x_{t-1}, c_{t-1})$ instead of $P(x_t|x_{t-1})$. Also note that X is independent of D_{t-1}, E_{t-1} given C_{t-1}, X_{t-1}.

Update to incorporate new evidence at time t:
$P\left(x_t|e_{0:t}, d_{0:t}, c_{0:t}\right) =$ **(v)** **(vi)** **(vii)** **Your choice for (i)**

(v)
- ○ $\left(P\left(c_t|c_{0:t-1}\right)\right)^{-1}$
- ● $\left(P\left(e_t, d_t, c_t|e_{0:t-1}, d_{0:t-1}, c_{0:t-1}\right)\right)^{-1}$
- ○ $\left(P\left(e_{0:t-1}, d_{0:t-1}, c_{0:t-1}|e_t, d_t, c_t\right)\right)^{-1}$
- ○ $\left(P\left(e_t|e_{0:t-1}\right) P\left(d_t|d_{0:t-1}\right) P\left(c_t|c_{0:t-1}\right)\right)^{-1}$
- ○ $\left(P\left(e_{0:t-1}|e_t\right) P\left(d_{0:t-1}|d_t\right) P\left(c_{0:t-1}|c_t\right)\right)^{-1}$
- ○ 1

(vi) ○ $\Sigma_{x_{t-1}}$ ○ Σ_{x_t} ○ Σ_{x_{t-1},x_t} ○ $\max_{x_{t-1}}$ ○ $\max_{x_t}$ ● 1

(vii)
- □ $P(x_t|e_t, d_t, c_t)$
- □ $P(x_t|e_t, d_t, c_t, c_{t-1})$
- ■ $P(e_t, d_t|x_t)P(c_t|e_t, d_t, c_{t-1})$
- □ $P(x_t, e_t, d_t, c_t)$
- □ $P(x_t, e_t, d_t, c_t, c_{t-1})$
- □ $P(e_t, d_t, c_t|x_t)$
- ○ 1

Recall the observation update of forward algorithm: $P(x_t|o_{0:t}) \propto P(x_t, o_t|o_{0:t-1}) = P(o_t|x_t)P(x_t|o_{0:t-1})$.
Here the observations o_t corresponds to e_t, d_t, c_t for each t. Apply the Chain Rule, we are having
$P\left(x_t|e_{0:t}, d_{0:t}, c_{0:t}\right) \propto P\left(x_t, e_t, d_t, c_t|e_{0:t-1}, d_{0:t-1}, c_{0:t-1}\right) = P(e_t, d_t, c_t|x_t, c_{t-1})P(x_t|e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$
$= P(e_t, d_t|x_t)P(c_t|e_t, d_t, c_{t-1})P(x_t|e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$.
Note that in $P(e_t, d_t, c_t|x_t, c_{t-1})$, we cannot omit c_{t-1} due to the arrow between c_t and c_{t-1}.
To calculate the normalizing constant, use Bayes Rule: $P\left(x_t|e_{0:t}, d_{0:t}, c_{0:t}\right) = \frac{P(x_t,e_t,d_t,c_t|e_{0:t-1},d_{0:t-1},c_{0:t-1})}{P(e_t,d_t,c_t|e_{0:t-1},d_{0:t-1},c_{0:t-1})}$.

(viii) Suppose we want to do the above updates in one step and use normalization to reduce computation. Select all the terms that are not explicitly calculated in this implementation.
DO **NOT** include the choices if their values are 1.

□ **(ii)** □ **(iii)** □ **(iv)** ■ **(v)** □ **(vi)** □ **(vii)** ○ None of the above

(v) is a constant, so we don't calculate it during implementation and simply do a normalization instead. Everything else is necessary.

(b) Suppose X outputs 1024×1024 greyscale images and our vehicle stays stationary. As before, E includes precise estimation of the distance between our vehicle and the pedestrian evaluated from outputs of X. Unfortunately, a power outage happened, and before the power is restored, E will not be available for our vehicle. But we still have the detection model D, which outputs one of {no pedestrian, pedestrian on the road, pedestrian beside the stop sign} for each state.

(i) During the power outage, it is best to
- ○ do particle filtering because the particles are easier to track for D than for both D and E
- ● do particle filtering because of memory constraints
- ○ do particle filtering, but not for the reasons above
- ○ do exact inference because it saves computation
- ○ do exact inference, but not for the reason above

E is unavailable and C does not change its value since our vehicle stays stationary, so we only considers X and D. Although D has only 3 possible values, X is huge and it is impossible to store the belief distribution.

(ii) The power outage was longer than expected. As the sensor outputs of X have degraded to 2×2 binary images, it is best to
- ○ do particle filtering because the particles are easier to track for D than for both D and E
- ○ do particle filtering because of memory constraints
- ○ do particle filtering, but not for the reasons above
- ○ do exact inference because it saves computation
- ● do exact inference, but not for the reason above

In this case we do not have the "X is huge" problem in **(i)**, and we can do exact inference, which is always more accurate than particle filtering and thus more favorable in this setting.

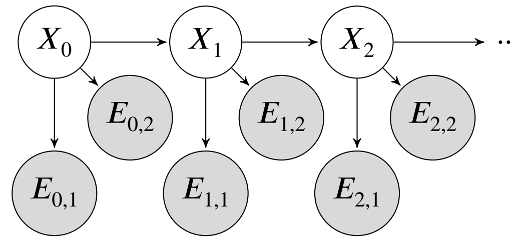
(iii) After power is restored and we have E, it is reasonable to
- ● do particle filtering because of memory constraints
- ○ do particle filtering, but not for the reason above
- ○ do exact inference because E gives more valuable information than D
- ○ do exact inference because it's impractical to do particle filtering for E
- ○ do exact inference, but not for the reasons above

The belief distribution is too big to store in memory.

Q2. Particle Filtering

You've chased your arch-nemesis Leland to the Stanford quad. You enlist two robo-watchmen to help find him! The grid below shows the campus, with ID numbers to label each region. Leland will be moving around the campus. His location at time step t will be represented by random variable X_t. Your robo-watchmen will also be on campus, but their locations will be fixed. Robot 1 is always in region 1 and robot 2 is always in region 9. (See the * locations on the map.) At each time step, each robot gives you a sensor reading to help you determine where Leland is. The sensor reading of robot 1 at time step t is represented by the random variable $E_{t,1}$. Similary, robot 2's sensor reading at time step t is $E_{t,2}$. The Bayes' Net to the right shows your model of Leland's location and your robots' sensor readings.

1*	2	3	4	5
6	7	8	9*	10
11	12	13	14	15



In each time step, Leland will either stay in the same region or move to an adjacent region. For example, the available actions from region 4 are (WEST, EAST, SOUTH, STAY). He chooses between all available actions with equal probability, regardless of where your robots are. Note: moving off the grid is not considered an available action.

Each robot will detect if Leland is in an adjacent region. For example, the regions adjacent to region 1 are 1, 2, and 6. If Leland is in an adjacent region, then the robot will report $NEAR$ with probability 0.8. If Leland is not in an adjacent region, then the robot will still report $NEAR$, but with probability 0.3.

For example, if Leland is in region 1 at time step t the probability tables are:

E	$P(E_{t,1} \mid X_t = 1)$	$P(E_{t,2} \mid X_t = 1)$
$NEAR$	0.8	0.3
FAR	0.2	0.7

(a) Suppose we are running particle filtering to track Leland's location, and we start at $t = 0$ with particles $[X = 6, X = 14, X = 9, X = 6]$. Apply a forward simulation update to each of the particles using the random numbers in the table below.

Assign region IDs to sample spaces in numerical order. For example, if, for a particular particle, there were three possible sucessor regions 10, 14 and 15, with associated probabilities, $P(X = 10)$, $P(X = 14)$ and $P(X = 15)$, and the random number was 0.6, then 10 should be selected if $0.6 \leq P(X = 10)$, 14 should be selected if $P(X = 10) < 0.6 < P(X = 10) + P(X = 14)$, and 15 should be selected otherwise.

Particle at $t = 0$	Random number for update	Particle after forward simulation update
$X = 6$	0.864	11
$X = 14$	0.178	9
$X = 9$	0.956	14
$X = 6$	0.790	11

(b) Some time passes and you now have particles [$X = 6$, $X = 1$, $X = 7$, $X = 8$] at the particular time step, but you have not yet incorporated your sensor readings at that time step. Your robots are still in regions 1 and 9, and both report $NEAR$. What weight do we assign to each particle in order to incorporate this evidence?

Particle	Weight
$X = 6$	$0.8 * 0.3$
$X = 1$	$0.8 * 0.3$
$X = 7$	$0.3 * 0.3$
$X = 8$	$0.3 * 0.8$

(c) To decouple this question from the previous question, let's say you just incorporated the sensor readings and found the following weights for each particle (these are not the correct answers to the previous problem!):

Particle	Weight
$X = 6$	0.1
$X = 1$	0.4
$X = 7$	0.1
$X = 8$	0.2

Normalizing gives us the distribution

$$X = 1 : 0.4/0.8 = 0.5$$
$$X = 6 : 0.1/0.8 = 0.125$$
$$X = 7 : 0.1/0.8 = 0.125$$
$$X = 8 : 0.2/0.8 = 0.25$$

Use the following random numbers to resample you particles. As on the previous page, **assign region IDs to sample spaces in numerical order.**

Random number:	0.596	0.289	0.058	0.765
Particle:	6	1	1	8