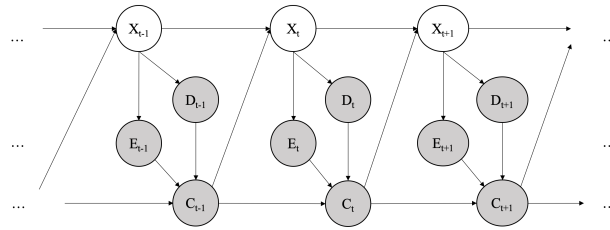


1 We Are Getting Close...

Mesut is trying to remotely control a car, which has gone out of his view. The unknown state of the car is represented by the random variable X . While Mesut can't see the car itself, his high-tech sensors on the car provides two useful readings: an estimate (E) of the distance to the car in front, and a detection model (D) that detects if the car is headed into a wall. Using these two readings, Mesut applies the controls (C), which determine the velocity of the car by changing the acceleration. The DBN below describes the setup.



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

Time elapse:	<u>(i)</u>	=	<u>(ii)</u>	<u>(iii)</u>	<u>(iv)</u>	$P(x_{t-1} e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$		
(i)	<input type="radio"/>	$P(x_t)$		<input type="radio"/>	$P(x_t e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$	<input type="radio"/>	$P(e_t, d_t, c_t e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$	
(ii)	<input type="radio"/>	$P(c_{0:t-1})$		<input type="radio"/>	$P(x_{0:t-1}, c_{0:t-1})$	<input type="radio"/>	$P(e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$	
	<input type="radio"/>	$P(e_{0:t}, d_{0:t}, c_{0:t})$		<input type="radio"/>	1			
(iii)	<input type="radio"/>	$\Sigma_{x_{t-1}}$	<input type="radio"/>	Σ_{x_t}	<input type="radio"/>	$\max_{x_{t-1}}$	<input type="radio"/>	\max_{x_t}
(iv)	<input type="radio"/>	$P(x_{t-1} x_{t-2})$		<input type="radio"/>	$P(x_{t-1}, x_{t-2})$	<input type="radio"/>	$P(x_t e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$	
	<input type="radio"/>	$P(x_t x_{t-1})$		<input type="radio"/>	$P(x_t, x_{t-1})$	<input type="radio"/>	$P(x_t, e_{0:t-1}, d_{0:t-1}, c_{0:t-1})$	
	<input type="radio"/>	$P(x_t x_{t-1}, c_{t-1})$		<input type="radio"/>	$P(x_t, x_{t-1}, c_{t-1})$	<input type="radio"/>	1	

Update to incorporate new evidence at time t :

$P(x_t e_{0:t}, d_{0:t}, c_{0:t}) =$		(v)	(vi)	(vii)	Your choice for (i)							
(v)	<input type="radio"/>	$(P(c_t c_{0:t-1}))^{-1}$			<input type="radio"/>	$(P(e_t e_{0:t-1}) P(d_t d_{0:t-1}) P(c_t c_{0:t-1}))^{-1}$						
	<input type="radio"/>	$(P(e_t, d_t, c_t e_{0:t-1}, d_{0:t-1}, c_{0:t-1}))^{-1}$			<input type="radio"/>	$(P(e_{0:t-1} e_t) P(d_{0:t-1} d_t) P(c_{0:t-1} c_t))^{-1}$						
	<input type="radio"/>	$(P(e_{0:t-1}, d_{0:t-1}, c_{0:t-1} e_t, d_t, c_t))^{-1}$			<input type="radio"/>	1						
(vi)	<input type="radio"/>	$\Sigma_{x_{t-1}}$	<input type="radio"/>	Σ_{x_t}	<input type="radio"/>	Σ_{x_{t-1}, x_t}	<input type="radio"/>	$\max_{x_{t-1}}$	<input type="radio"/>	\max_{x_t}	<input type="radio"/>	1
(vii)	<input type="checkbox"/>	$P(x_t e_t, d_t, c_t)$			<input type="checkbox"/>	$P(x_t, e_t, d_t, c_t)$						
	<input type="checkbox"/>	$P(x_t e_t, d_t, c_t, c_{t-1})$			<input type="checkbox"/>	$P(x_t, e_t, d_t, c_t, c_{t-1})$						
	<input type="checkbox"/>	$P(e_t, d_t x_t)P(c_t e_t, d_t, c_{t-1})$			<input type="checkbox"/>	$P(e_t, d_t, c_t x_t)$					<input type="radio"/>	1

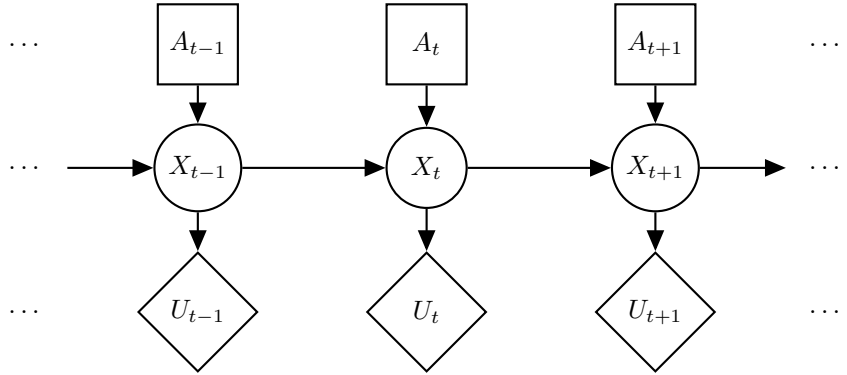
(viii) Suppose we want to do the above updates in one step and use normalization to reduce computation. Select all the terms that are not explicitly calculated in this implementation.

DO NOT include the choices if their values are 1.

- ☐ (ii) ☐ (iii) ☐ (iv) ☐ (v) ☐ (vi) ☐ (vii) ☐ None of the above

2 Planning Ahead with HMMs

Pacman is tired of using HMMs to estimate the location of ghosts. He wants to use HMMs to plan what actions to take in order to maximize his utility. Pacman uses the HMM (drawn to the right) of length T to model the planning problem. In the HMM, $X_{1:T}$ is the sequence of hidden states of Pacman's world, $A_{1:T}$ are actions Pacman can take, and U_t is the utility Pacman receives at the particular hidden state X_t . Notice that there are no evidence variables, and utilities are not discounted.



- (a) The belief at time t is defined as $B_t(X_t) = p(X_t|a_{1:t})$. The forward algorithm update has the following form:

$$B_t(X_t) = \underline{\hspace{1cm}} \text{ (i) } \underline{\hspace{1cm}} \text{ (ii) } \underline{\hspace{1cm}} B_{t-1}(x_{t-1}).$$

Complete the expression by choosing the option that fills in each blank.

- (i) ☐ $\max_{x_{t-1}}$ ☐ $\sum_{x_{t-1}}$ ☐ \max_{x_t} ☐ \sum_{x_t} ☐ 1
(ii) ☐ $p(X_t|x_{t-1})$ ☐ $p(X_t|x_{t-1})p(X_t|a_t)$ ☐ $p(X_t)$ ☐ $p(X_t|x_{t-1}, a_t)$ ☐ 1
☐ None of the above combinations is correct

- (b) Pacman would like to take actions $A_{1:T}$ that maximizes the expected sum of utilities, which has the following form:

$$\text{MEU}_{1:T} = \underline{\hspace{1cm}} \text{ (i) } \underline{\hspace{1cm}} \text{ (ii) } \underline{\hspace{1cm}} \text{ (iii) } \underline{\hspace{1cm}} \text{ (iv) } \underline{\hspace{1cm}} \text{ (v) }$$

Complete the expression by choosing the option that fills in each blank.

- (i) ☐ $\max_{a_{1:T}}$ ☐ \max_{a_T} ☐ $\sum_{a_{1:T}}$ ☐ \sum_{a_T} ☐ 1
(ii) ☐ \max_t ☐ $\prod_{t=1}^T$ ☐ $\sum_{t=1}^T$ ☐ \min_t ☐ 1
(iii) ☐ \sum_{x_t, a_t} ☐ \sum_{x_t} ☐ \sum_{a_t} ☐ \sum_{x_T} ☐ 1
(iv) ☐ $p(x_t|x_{t-1}, a_t)$ ☐ $p(x_t)$ ☐ $B_t(x_t)$ ☐ $B_T(x_T)$ ☐ 1
(v) ☐ U_T ☐ $\frac{1}{U_t}$ ☐ $\frac{1}{U_T}$ ☐ U_t ☐ 1
☐ None of the above combinations is correct

- (c) A greedy ghost now offers to tell Pacman the values of some of the hidden states. Pacman needs your help to figure out if the ghost's information is useful. Assume that the transition function $p(x_t|x_{t-1}, a_t)$ is not deterministic. **With respect to the utility U_t** , mark all that can be True:

- ☐ $\text{VPI}(X_{t-1}|X_{t-2}) > 0$ ☐ $\text{VPI}(X_{t-2}|X_{t-1}) > 0$ ☐ $\text{VPI}(X_{t-1}|X_{t-2}) = 0$ ☐ $\text{VPI}(X_{t-2}|X_{t-1}) = 0$
☐ None of the above

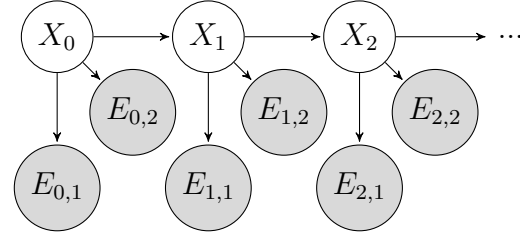
- (d) Pacman notices that calculating the beliefs under this model is very slow using exact inference. He therefore decides to try out various particle filter methods to speed up inference. Order the following methods by how accurate their estimate of $B_T(X_T)$ is? If different methods give an equivalently accurate estimate, mark them as the same number.

	Most accurate			Least accurate		
Exact inference	○ 1	○ 2	○ 3	○ 4	○ 4	○ 4
Particle filtering with no resampling	○ 1	○ 2	○ 3	○ 4	○ 4	○ 4
Particle filtering with resampling before every time elapse	○ 1	○ 2	○ 3	○ 4	○ 4	○ 4
Particle filtering with resampling before every other time elapse	○ 1	○ 2	○ 3	○ 4	○ 4	○ 4

3 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 $E_{t,1}$. Similarly, 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	$P(E_{t,1} X_t = 1)$	$P(E_{t,2} X_t = 1)$
<i>NEAR</i>	0.8	0.3
<i>FAR</i>	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 successor 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	
$X = 14$	0.178	
$X = 9$	0.956	
$X = 6$	0.790	

- (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
$X = 6$	
$X = 1$	
$X = 7$	
$X = 8$	

- (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

Use the following random numbers to resample your 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:				