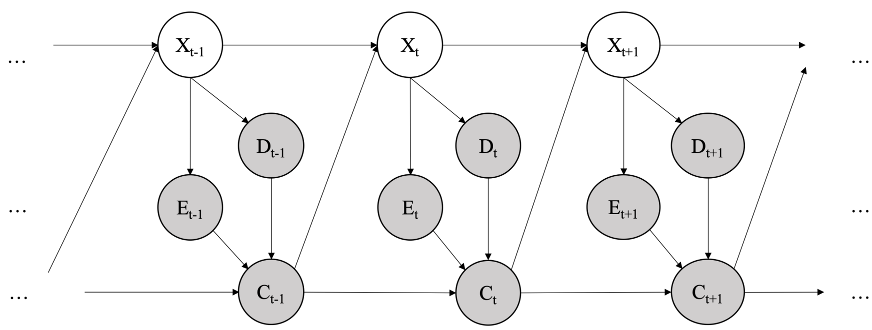
# Q1. We Are Getting Close...

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 Dynamic Bayes Net 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: **(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)** ● $\Sigma_{x_{t-1}}$ ○ $\Sigma_{x_t}$ ○ $\max_{x_{t-1}}$ ○ $\max_{x_t}$ ○ 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)** ○ $\left(P\left(c_t|c_{0:t-1}\right)\right)^{-1}$ ○ $\left(P\left(e_t|e_{0:t-1}\right) P\left(d_t|d_{0:t-1}\right) P\left(c_t|c_{0:t-1}\right)\right)^{-1}$
● $\left(P\left(e_t, d_t, c_t|e_{0:t-1}, d_{0:t-1}, c_{0:t-1}\right)\right)^{-1}$ ○ $\left(P\left(e_{0:t-1}|e_t\right) P\left(d_{0:t-1}|d_t\right) P\left(c_{0:t-1}|c_t\right)\right)^{-1}$
○ $\left(P\left(e_{0:t-1}, d_{0:t-1}, c_{0:t-1}|e_t, d_t, c_t\right)\right)^{-1}$ ○ 1

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

**(vii)** ☐ $P(x_t|e_t, d_t, c_t)$ ☐ $P(x_t, e_t, d_t, c_t)$
☐ $P(x_t|e_t, d_t, c_t, c_{t-1})$ ☐ $P(x_t, e_t, d_t, c_t, c_{t-1})$
■ $P(e_t, d_t|x_t)P(c_t|e_t, d_t, c_{t-1})$ ☐ $P(e_t, d_t, c_t|x_t)$ ○ 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: $P\left(x_t|e_{0:t}, d_{0:t}, c_{0:t}\right) = \frac{P(x_t,e_t,d_t,c_t|e_{0:t-1},d_{0:t-1},c_{0:t-1})}{P(e_t,d_t,c_t|e_{0:t-1},d_{0:t-1},c_{0:t-1})}$.

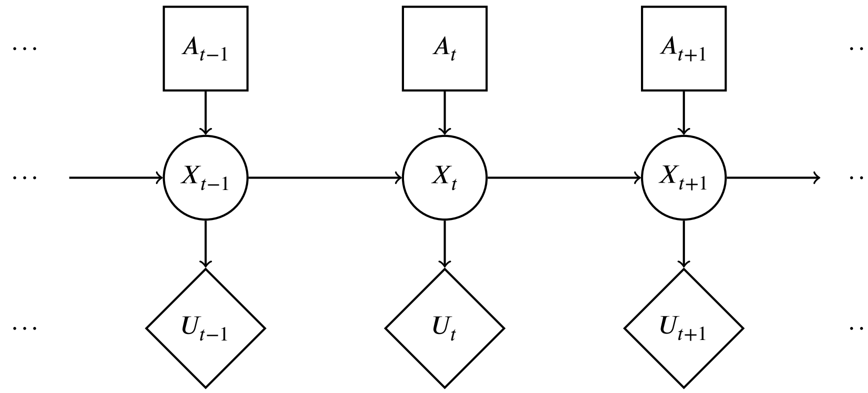
**(viii)** Suppose we want to do the above updates in one step and use normalization to reduce computation. Select all the terms that are not explicitly calculated in this implementation.
DO **NOT** include the choices if their values are 1.

☐ **(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.

# Q2. 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:

$$B_t(X_t) = \underline{\quad\textbf{(i)}\quad}\ \underline{\quad\textbf{(ii)}\quad}\ B_{t-1}(x_{t-1}).$$

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

$$\begin{aligned} B_t(X_t) &= p(X_t|a_{1:t}) \\ &= \sum_{x_{t-1}} p(X_t|x_{t-1}, a_t)p(x_{t-1}|a_{1:t-1}) \\ &= \sum_{x_{t-1}} p(X_t|x_{t-1}, a_t)B_{t-1}(x_{t-1}) \end{aligned}$$

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

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

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

$$\text{MEU}_{1:T} = \max_{a_{1:T}} \sum_{t=1}^{T} \sum_{x_t} B_t(x_t)U_t(x_t)$$

**(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:

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

It is always possible that VPI $= 0$. Can guarantee VPI($E|e$) is not greater than 0 if $E$ is independent of parents($U$) given $e$.

**(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 |
| Particle filtering with no resampling | ○ 1 | ● 2 | ○ 3 | ○ 4 |
| Particle filtering with resampling before every time elapse | ○ 1 | ○ 2 | ○ 3 | ● 4 |
| Particle filtering with resampling before every other time elapse | ○ 1 | ○ 2 | ● 3 | ○ 4 |

Exact inference will always be more accurate than using a particle filter. When comparing the particle filter resampling approaches, notice that because there are no observations, each particle will have weight 1. Therefore resampling when particle weights are 1 could lead to particles being lost and hence prove bad.