COMP 341 Intro to AI
Bayesian Networks – Reasoning Over Time

Asst. Prof. Barış Akgün

Koç University

Recap

- Uncertainty
  - The real world is uncertain to an agent!
  - Use probabilistic models for representation – Joint Distribution
- Bayesian Networks
  - An intuitive way of representing uncertainty with local conditional distributions
- Inference in BNs: $P(X_q | x_{e_1}, \ldots x_{e_k})$

  Stuff you care about   *Stuff you already know*
- Exact Inference: Enumeration
- Approximate Inference: Sampling
- What to do with the inference outcome? Decision Networks
- Is it worth it to collect evidence? Value of Information

Reasoning Over Time

- When we want to *reason about a sequence* of observations
- So why or when would we want this?

- Need to introduce time, sequencing or dynamics into our models
- Basic Approach: Hidden Markov Models (HMMs)
- More general: Dynamic Bayes Nets (DBNs)

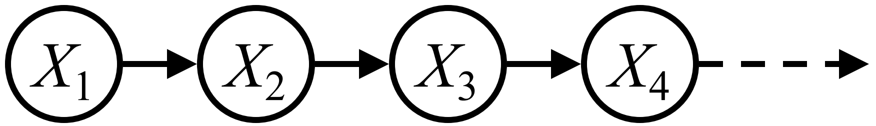
Some Applications of Reasoning Over Time

- Speech recognition and synthesis
- Cryptanalysis
- Financial Market Analysis
- Activity Recognition
- Sequence Alignment (not reasoning over time but analogous, e.g. gene-sequences)
- Signature/Sketch Recognition
- Robot Localization
- Monitoring (e.g. medical applications)

…

Markov Models

- We are going to cover Markov models in the context of BNs
    - There are other treatments of the subject but most of them are for stochastic systems
- A Markov model is a chain structured BN



$P(X_1)$ $\quad$ $P(X_t|X_{t-1})$

- Value of X at a given time is called the state
- Parameters:
    - Transition Probabilities (or Dynamics): How state evolves over time $P(X_t|X_{t-1})$
    - Initial State Probabilities (or Prior Probabilities): Probability of a state being the first state $P(X_t)$
- Stationarity assumption: transition probabilities do not change over time

Detour: Markov Property

- The conditional probability of the next state only depends on the current state

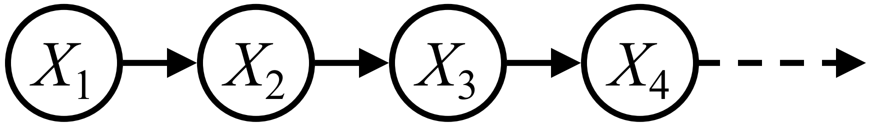
$$P(X_{t+1}|X_t, X_{t-1}, \dots, X_1) = P(X_{t+1}|X_t)$$

- Markov **Assumption**: When we assume that a model has the Markov property
- Controlled systems (e.g. robots) where we can have control inputs, we can also apply the Markov assumption:

$P(X_{t+1}|X_t, U_t)$, U_t is the control input

- This implies that the next state depends only on the current state and the current control input

Conditional Independence in Markov Models



$P(X_1)$ $\quad$ $P(X_t|X_{t-1})$

- Basic conditional independence:
  - (First order) Markov property
  - Past and future independent of the present
  - Each time step only depends on the previous

- Note that the chain is just a (growing) BN
  - We can always use generic BN reasoning on it if we truncate the chain at a fixed length

Joint Distribution of a Markov Model

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4$$

$P(X_1)$ $P(X_t|X_{t-1})$

- Joint Distribution

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_3)$$

- In general

$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2|X_1) \ldots P(X_T|X_{T-1})$$
$$= P(X_1) \prod_{t=2}^{T} P(X_t|X_{t-1})$$

Higher Order Markov Models

- $P(X_t|X_{t-1})$
- $P(X_t|X_{t-1}, X_{t-2})$
- $P(X_t|X_{t-1}, X_{t-2}, .., X_{t-\omega})$

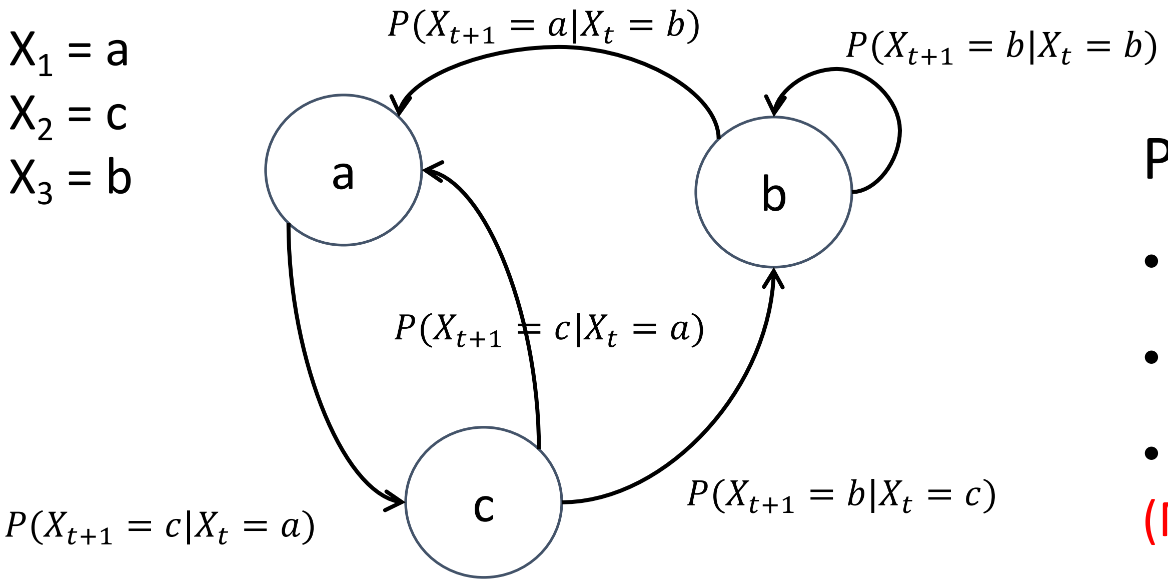
  $\Rightarrow P(\bar{X}_t|\bar{X}_{t-1})$ where $\bar{X}_{t-1} = [X_{t-1}, X_{t-2}, .., X_{t-\omega}]$

- $P(X_t|X_{t-1}, X_{t-2}, .., X_0)$?

Markov Chains

Discrete state – Discrete time Random Dynamical System

X_1 = a
X_2 = c
X_3 = b



Properties

- Finite number of states (N) (for this example a, b and c)
- State transitions are random
- The next state only depends on the current state (Markov Assumption)
- States are directly observable

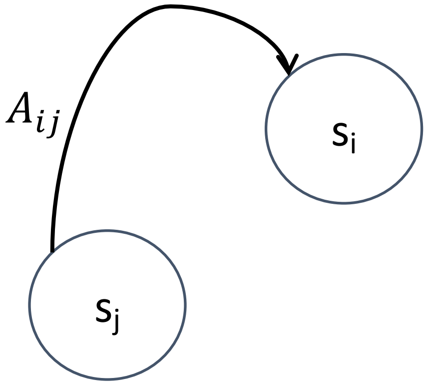
Note that this describes a conditional probability table ($P(X_{t+1}|X_t)$), not a Bayesian Network!

Markov Chains

- Transition Matrix T:

$$T_{ij} = P(X_{t+1} = s_i | X_t = s_j)$$

t+1 \ t	s_1	s_2	s_3
s_1	$P(X_{t+1} = s_1 \mid X_t = s_1)$	$P(X_{t+1} = s_1 \mid X_t = s_2)$	$P(X_{t+1} = s_1 \mid X_t = s_3)$
s_2	$P(X_{t+1} = s_2 \mid X_t = s_1)$	$P(X_{t+1} = s_2 \mid X_t = s_2)$	$P(X_{t+1} = s_2 \mid X_t = s_3)$
s_3	$P(X_{t+1} = s_3 \mid X_t = s_1)$	$P(X_{t+1} = s_3 \mid X_t = s_2)$	$P(X_{t+1} = s_3 \mid X_t = s_3)$



- Prior Probabilities:

$$\rho = [P(X_1 = s_1), \ldots, P(X_1 = s_N)]$$

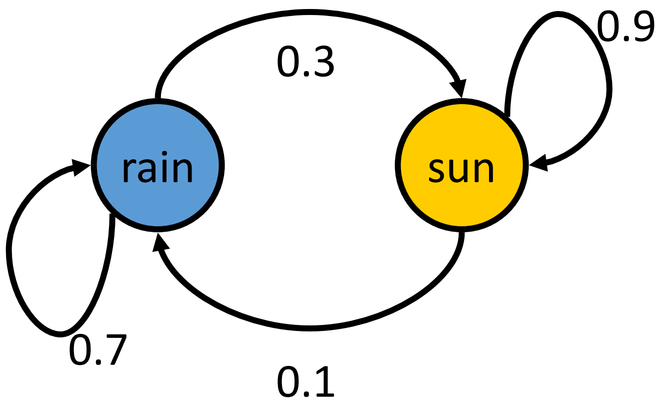
Example Markov Chain: Weather

- States: X = {rain, sun}

- Initial distribution: 1.0 sun

- CPT $P(X_t \mid X_{t-1})$:

X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7



What's the probability distribution after one step?

$$P(X_2 = sun) = ?$$

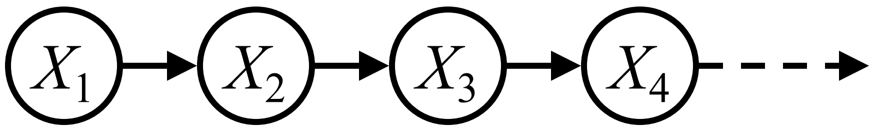
$$P(X_2 = sun) = P(X_2 = sun \mid X_1 = sun)P(X_1 = sun)$$
$$+ P(X_2 = sun \mid X_1 = rain)P(X_1 = rain)$$
$$= 0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9$$

What about these?

$$P(X_2 = rain) = ?, P(X_3 = sun) = ?$$

Mini-Forward Algorithm

- Question: What's P(X) on some day t?



$P(x_1) =$ known

$$P(x_t) = \sum_{t-1} P(x_{t-1}, x_t)$$

$$= \sum_{t-1} P(x_t | x_{t-1}) P(x_{t-1})$$

Forward simulation

Linear Algebra Equivalent

$P(x_1) =$ known

$$P(x_t) = \sum_{t-1} P(x_{t-1}, x_t)$$

$$= \sum_{t-1} P(x_t | x_{t-1}) P(x_{t-1})$$

$T_{ij} = P(X_{t+1} = s_i | X_t = s_j)$, $\rho = [P(X_1 = s_1), \dots, P(X_1 = s_N)]^T$

t+1 \ t	s_1	S_2	s_3
s_1	$P(X_{t+1} = s_1 \mid X_t = s_1)$	$P(X_{t+1} = s_1 \mid X_t = s_2)$	$P(X_{t+1} = s_1 \mid X_t = s_3)$
s_2	$P(X_{t+1} = s_2 \mid X_t = s_1)$	$P(X_{t+1} = s_2 \mid X_t = s_2)$	$P(X_{t+1} = s_2 \mid X_t = s_3)$
s_3	$P(X_{t+1} = s_3 \mid X_t = s_1)$	$P(X_{t+1} = s_3 \mid X_t = s_2)$	$P(X_{t+1} = s_3 \mid X_t = s_3)$

Let $P_t = [P(X_t = s_1), \dots, P(X_t = s_N)]$ ($P_1 = \rho$, known)

$$P(X_2 = s_1) = T_{11} P_1(X_1 = s_1) + T_{12} P_1(X_1 = s_2) + \cdots + T_{1N} P_1(X_1 = s_N) = [\mathrm{T}_{11}, \mathrm{T}_{12}, \dots, \mathrm{T}_{1\mathrm{N}}] \cdot P_1$$

$$\Rightarrow P_t = T P_{t-1}$$

$$P_t = T^{t-1} P_1$$

Example Run of Mini-Forward Algorithm

- From initial observation of sun

$$\underset{P(X_1)}{\left\langle \begin{matrix} 1.0 \\ 0.0 \end{matrix} \right\rangle} \quad \underset{P(X_2)}{\left\langle \begin{matrix} 0.9 \\ 0.1 \end{matrix} \right\rangle} \quad \underset{P(X_3)}{\left\langle \begin{matrix} 0.84 \\ 0.16 \end{matrix} \right\rangle} \quad \underset{P(X_4)}{\left\langle \begin{matrix} 0.804 \\ 0.196 \end{matrix} \right\rangle} \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{matrix} 0.75 \\ 0.25 \end{matrix} \right\rangle}$$

- From initial observation of rain

$$\underset{P(X_1)}{\left\langle \begin{matrix} 0.0 \\ 1.0 \end{matrix} \right\rangle} \quad \underset{P(X_2)}{\left\langle \begin{matrix} 0.3 \\ 0.7 \end{matrix} \right\rangle} \quad \underset{P(X_3)}{\left\langle \begin{matrix} 0.48 \\ 0.52 \end{matrix} \right\rangle} \quad \underset{P(X_4)}{\left\langle \begin{matrix} 0.588 \\ 0.412 \end{matrix} \right\rangle} \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{matrix} 0.75 \\ 0.25 \end{matrix} \right\rangle}$$

- From yet another initial distribution $P(X_1)$:

$$\underset{P(X_1)}{\left\langle \begin{matrix} p \\ 1-p \end{matrix} \right\rangle} \quad \ldots \quad \Rightarrow \underset{P(X_\infty)}{\left\langle \begin{matrix} 0.75 \\ 0.25 \end{matrix} \right\rangle}$$

Stationary Distributions

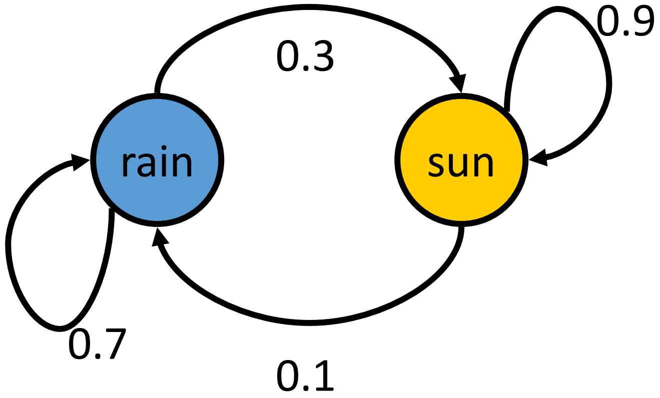
- If we simulate the chain long enough:
  - What happens?
  - Uncertainty accumulates
  - Eventually, we have no idea what the state is!

- Stationary distributions:
  - For most chains, the distribution we end up in is independent of the initial distribution
  - Called the stationary distribution of the chain
  - Usually, can only predict a short time out

Example: Stationary Distributions

- Question: What's P(X) at time t = infinity?

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4 \dashrightarrow$$



$$P_\infty(sun) = P(sun|sun)P_\infty(sun) + P(sun|rain)P_\infty(rain)$$
$$P_\infty(rain) = P(rain|sun)P_\infty(sun) + P(rain|rain)P_\infty(rain)$$

$$P_\infty(sun) = 0.9P_\infty(sun) + 0.3P_\infty(rain)$$
$$P_\infty(rain) = 0.1P_\infty(sun) + 0.7P_\infty(rain)$$

$$P_\infty(sun) = 3P_\infty(rain)$$
$$P_\infty(rain) = 1/3P_\infty(sun)$$

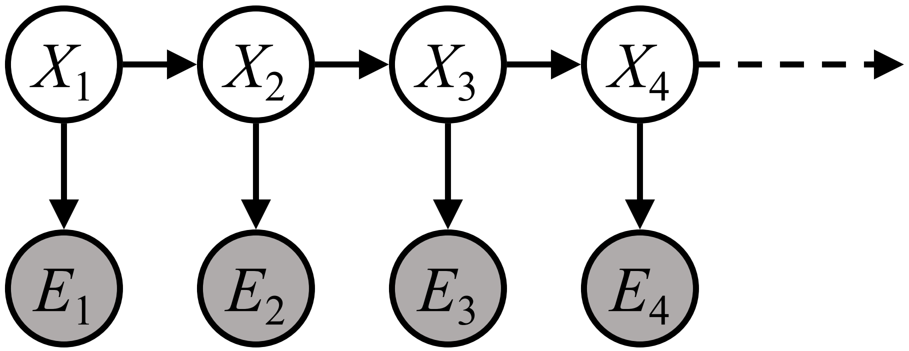
Also: $P_\infty(sun) + P_\infty(rain) = 1$ $\Rightarrow$

$$P_\infty(sun) = 3/4$$
$$P_\infty(rain) = 1/4$$

Hidden Markov Models

- Markov chains are not so useful for most agents
  - Forward simulating: Eventually you don't know anything anymore
  - Need observations to update beliefs
  - Cannot observe the states directly (e.g. measurement uncertainty)

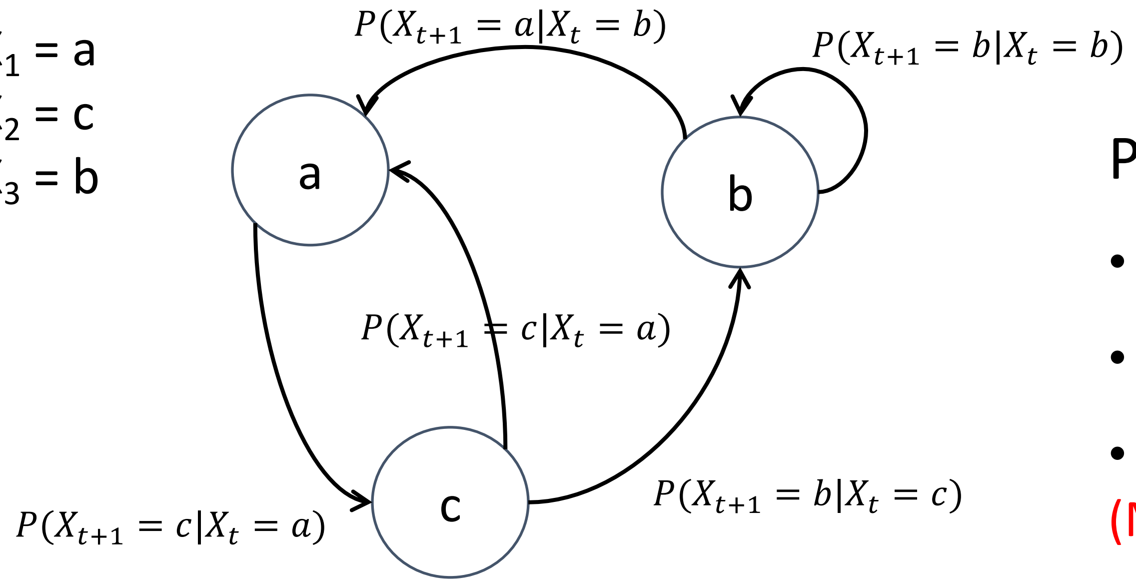
- Hidden Markov Models(HMMs)
  - Underlying Markov chain over states X
  - Observe outputs (Effects) at each time step



Markov Chains

Discrete state – Discrete time Random Dynamical System

$X_1 = a$
$X_2 = c$
$X_3 = b$



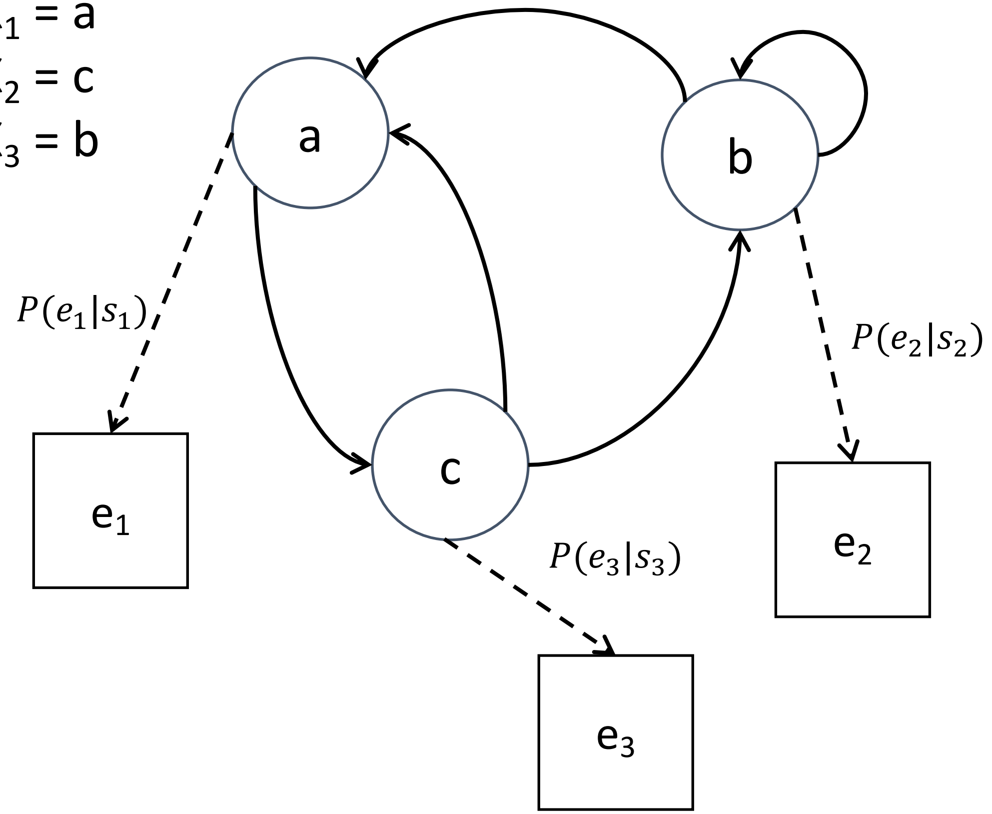
Properties

- Finite number of states (N) (for this example a, b and c)
- State transitions are random
- The next state only depends on the current state (Markov Assumption)
- States are directly observable

Markov Chains

Discrete state – Discrete time Random Dynamical System

$X_1 = a$
$X_2 = c$
$X_3 = b$



Properties

- Finite number of states (N) (for this example a, b and c)
- State transitions are random
- The next state only depends on the current state (Markov Assumption)
- ~~States are directly observable~~
- States are observed via a noisy process

Hidden Markov Models

- Transition Model $P(X_{t+1}|X_t)$:

$$A_{ij} = P(X_{t+1} = s_i | X_t = s_j)$$

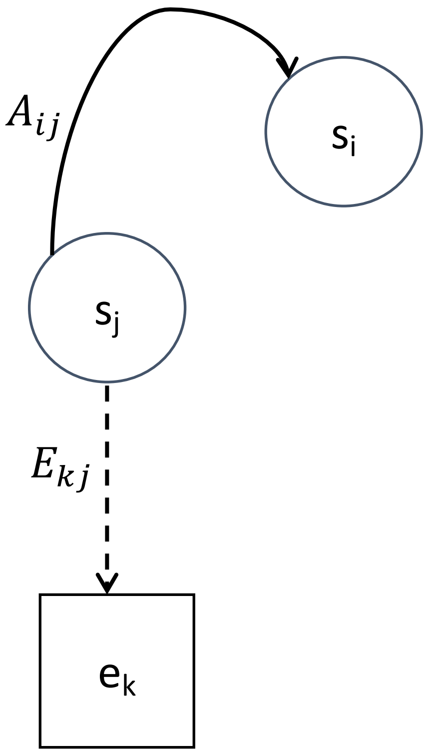
- Prior Probabilities $P(X_t)$:

$$\rho = [P(X_1 = s_1), \ldots, P(X_1 = s_N)]$$

- Emission Model $P(E_t|X_t)$:

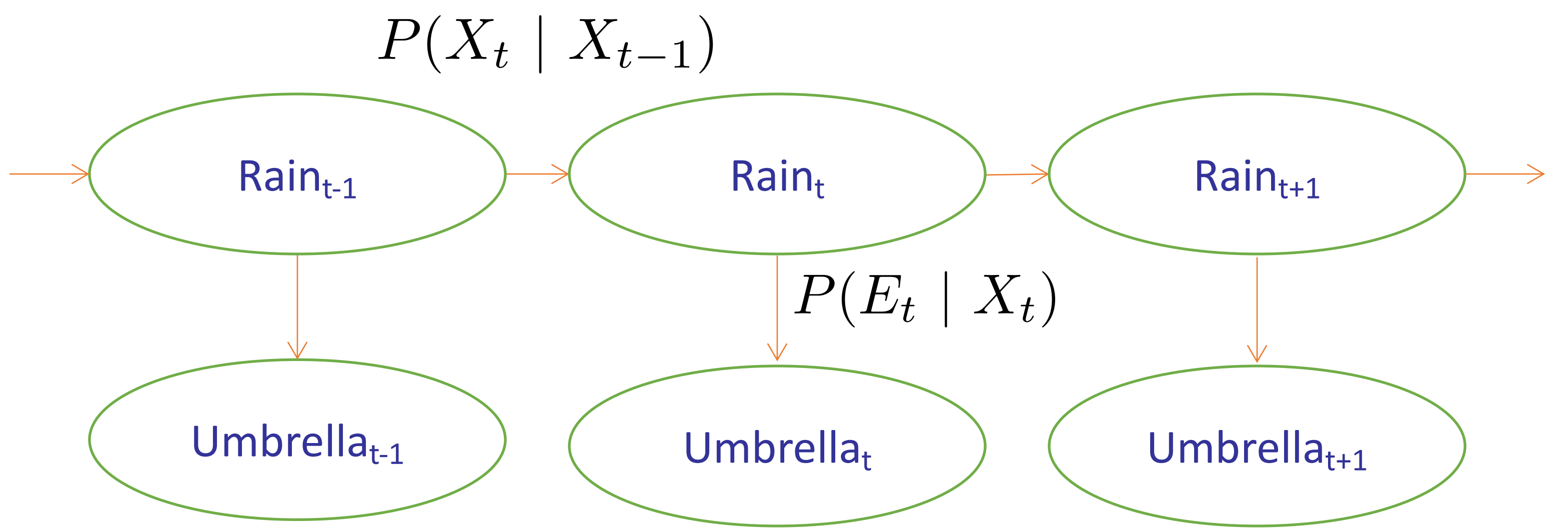
$$E_{kj} = P(E_t = e_k | X_t = s_i)$$

t / t+1	s_1	s_2	s_3
s_1	$P(X_{t+1} = s_1 \mid X_t = s_1)$	$P(X_{t+1} = s_1 \mid X_t = s_2)$	$P(X_{t+1} = s_1 \mid X_t = s_3)$
s_2	$P(X_{t+1} = s_2 \mid X_t = s_1)$	$P(X_{t+1} = s_2 \mid X_t = s_2)$	$P(X_{t+1} = s_2 \mid X_t = s_3)$
s_3	$P(X_{t+1} = s_3 \mid X_t = s_1)$	$P(X_{t+1} = s_3 \mid X_t = s_2)$	$P(X_{t+1} = s_3 \mid X_t = s_3)$



E / X	s_1	s_2	s_3
e_1	$P(E_t = e_1 \mid X_t = s_1)$	$P(E_t = e_! \mid X_t = s_2)$	$P(E_t = e_1 \mid X_t = s_3)$
e_2	$P(E_t = e_2 \mid X_t = s_1)$	$P(E_t = e_2 \mid X_t = s_2)$	$P(E_t = e_2 \mid X_t = s_3)$
e_3	$P(E_t = e_3 \mid X_t = s_1)$	$P(E_t = e_3 \mid X_t = s_2)$	$P(E_t = e_3 \mid X_t = s_3)$

Example: Weather HMM

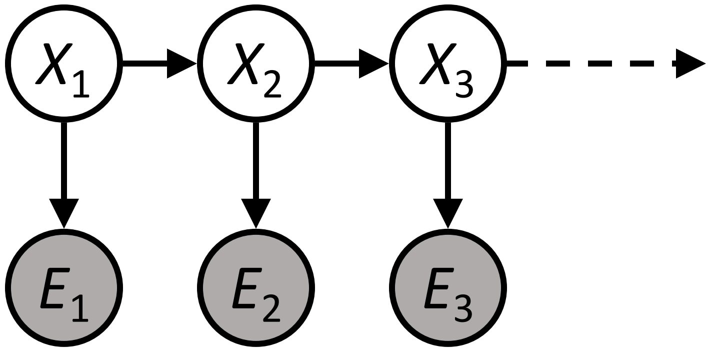


- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t \mid X_{t-1})$
  - Emissions: $P(E_t \mid X_t)$

R_t	R_{t+1}	$P(R_{t+1} \mid R_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

Joint Distribution of an HMM



- Joint Distribution

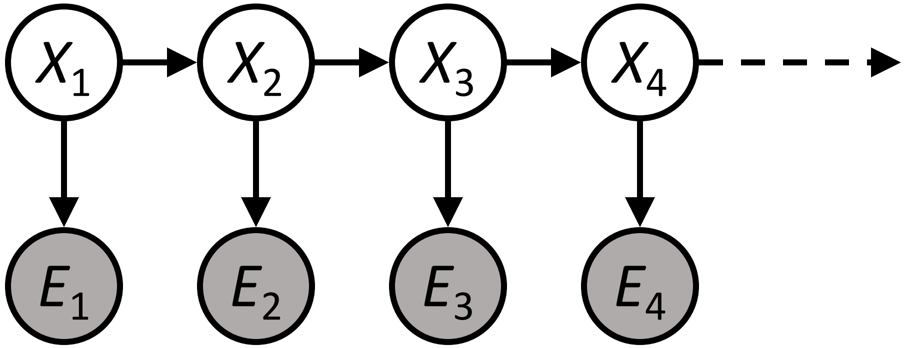
$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3)$$

- In general

$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t)$$

Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process: future depends on past via the present
  - Current observations/effects/emissions independent of all else given current state



Real HMM Examples

- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)

- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options

- Robot tracking:
  - Observations are range readings (continuous – can be discretized)
  - States are positions on a map (continuous – can be discretized)

What to do with HMMs?

- Keep track of “something” over time (Filtering)
- Most likely explanation (or history)
- Likelihood of an observed sequence
- Re-analyzing measurements (Smoothing)

Filtering/Monitoring

- Filtering or monitoring is the task of tracking the distribution $B_t(X) = P_t(X_t | e_1, \ldots, e_t)$, called the belief state, over time

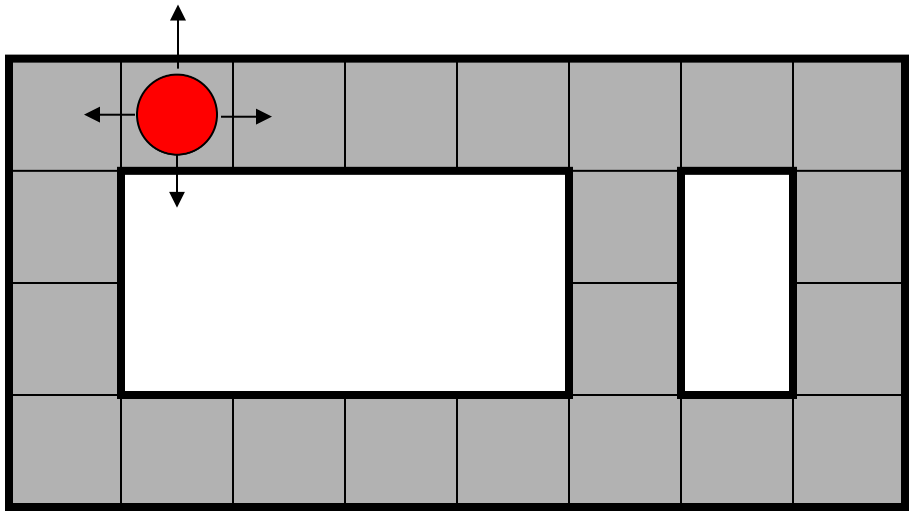
- Start with $B_1(X)$, can be uniform
- Update $\mathrm{B}(X)$, as time passes, or we get more evidence
  - Time passing: State evolves from X_{t-1} to X_t
  - More evidence: Get an emission E_t at X_t

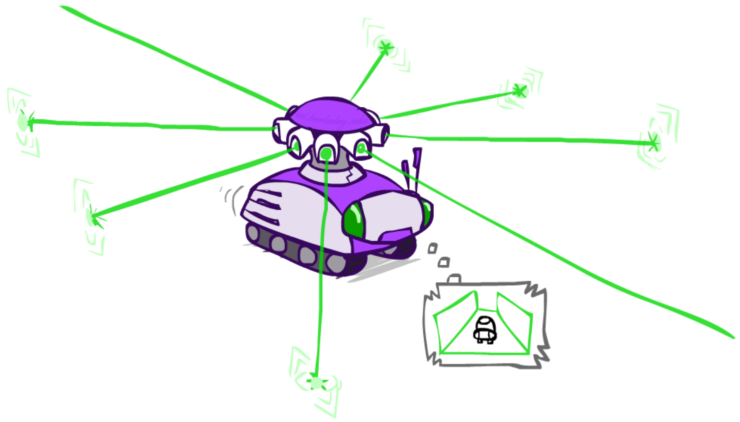
- The Kalman filter was invented in the 60's and first implemented as a method of trajectory estimation for the Apollo program

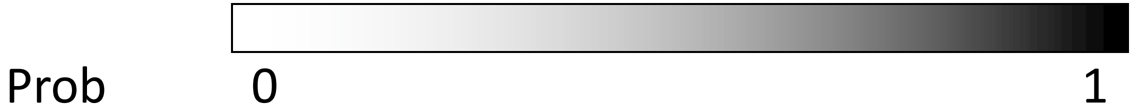
Example: Robot Localization

X: Tiles
E: 4 sensor readings

Example from Michael Pfeiffer







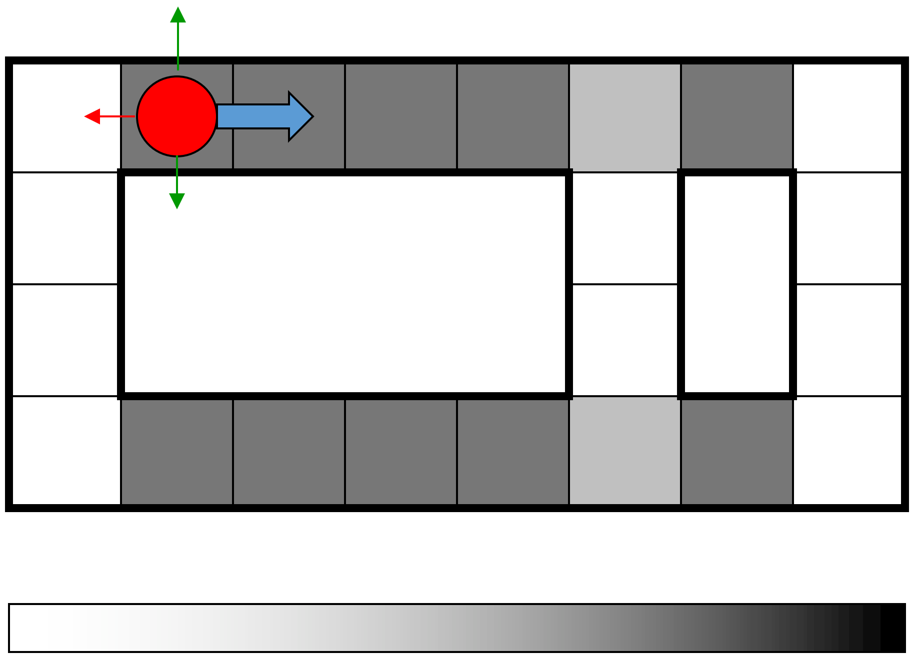
t=0

Sensor model: can read in which directions there is a wall, never more than 1 mistake

Motion model: may not execute action with small prob.

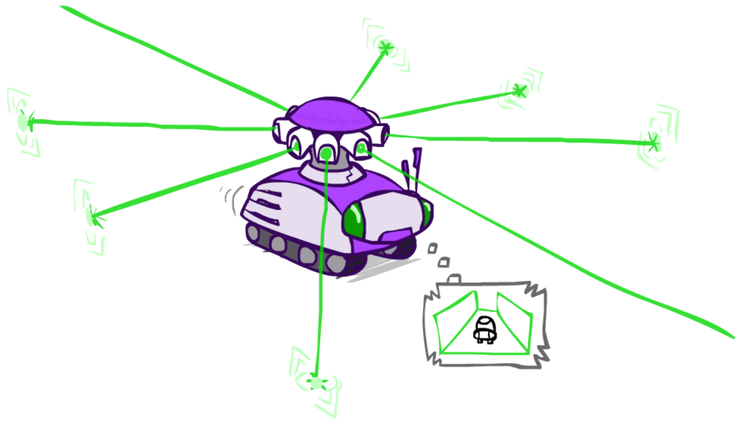
Example: Robot Localization

X: Tiles
E: 4 sensor readings



Prob 0 1

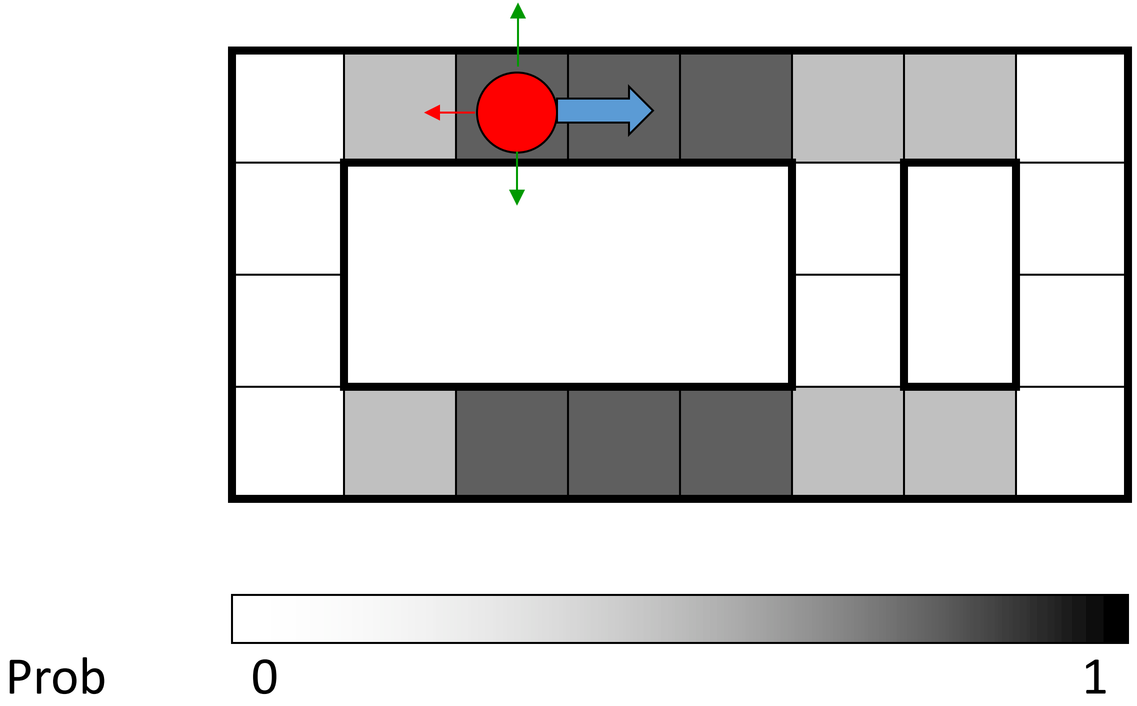
t=1



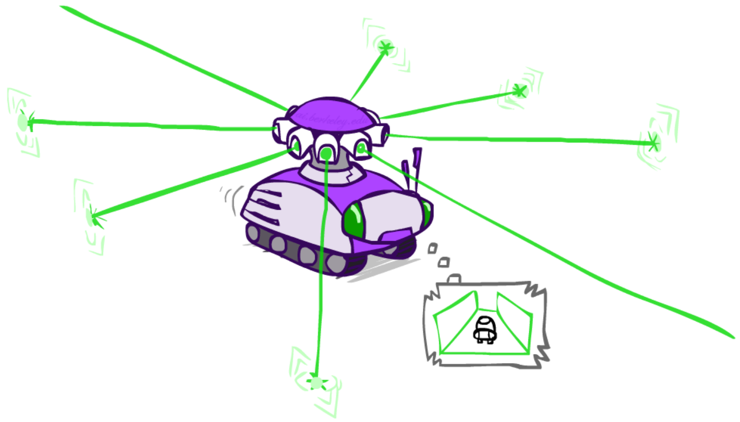
Lighter grey: was possible to get the reading, but less likely b/c required 1 mistake

Example: Robot Localization

X: Tiles
E: 4 sensor readings

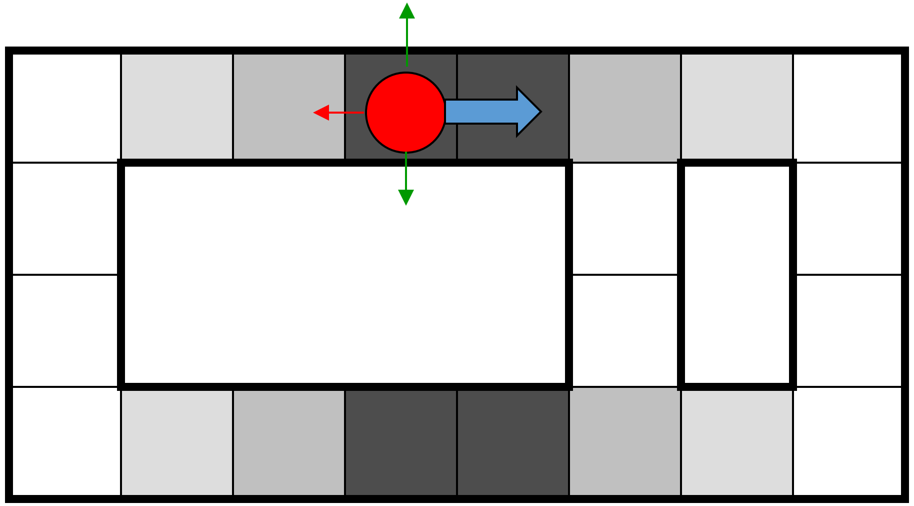


t=2



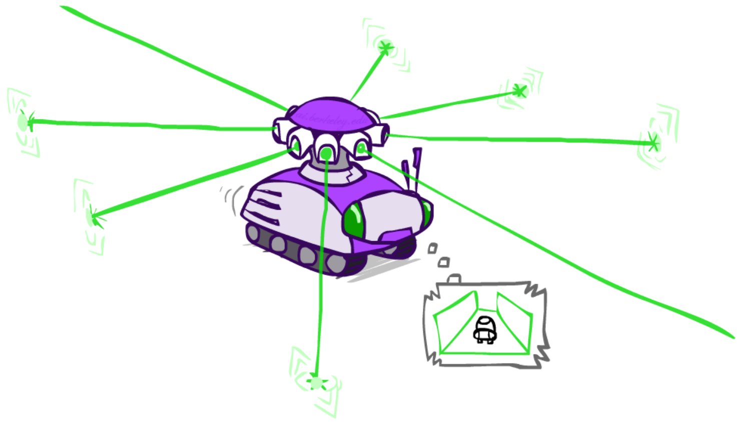
Example: Robot Localization

X: Tiles
E: 4 sensor readings



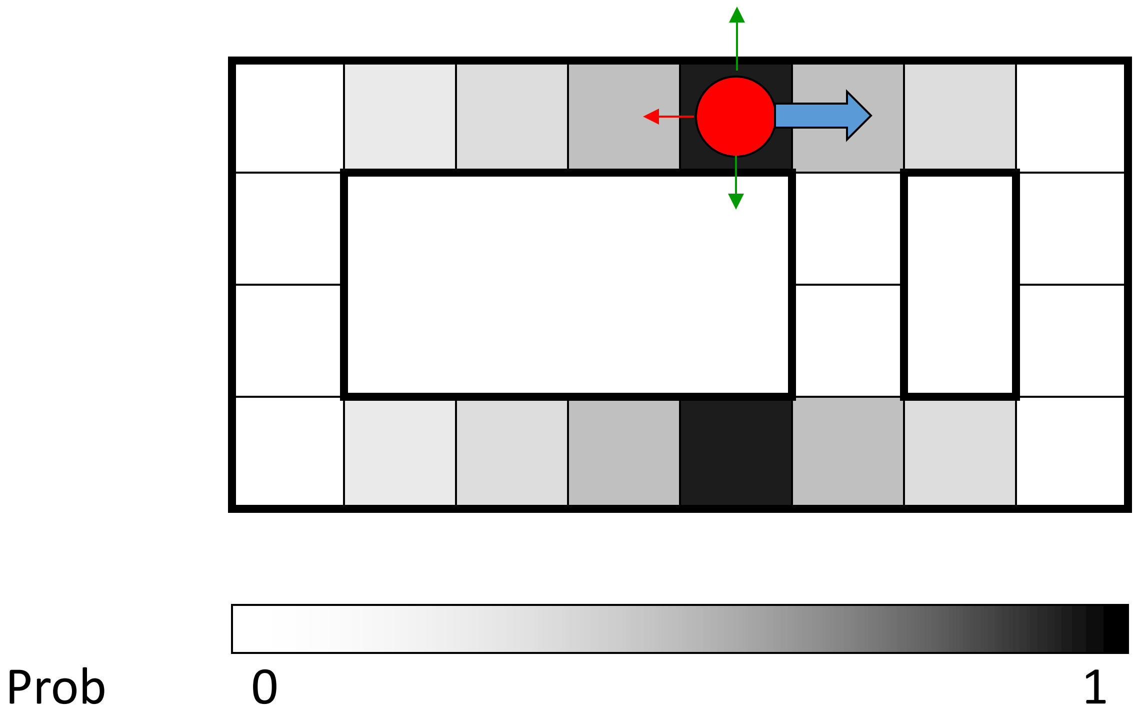
Prob 0 1

t=3

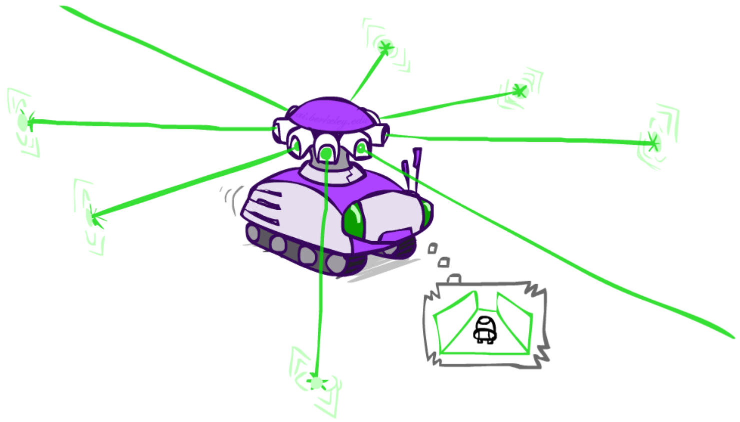


Example: Robot Localization

X: Tiles
E: 4 sensor readings

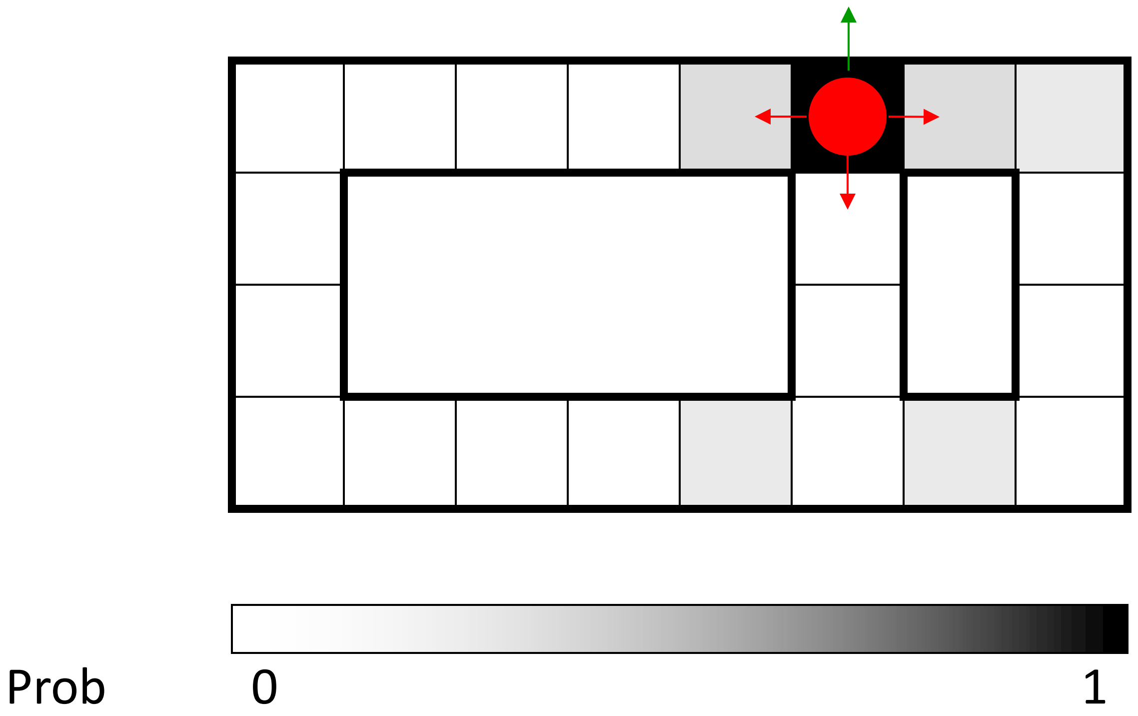


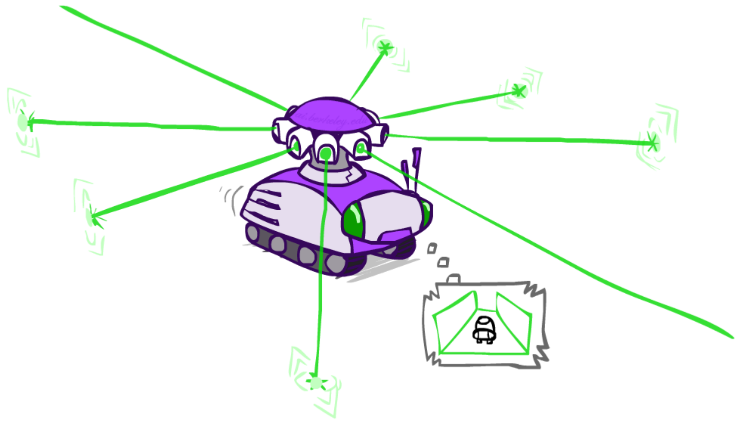
t=4



Example: Robot Localization

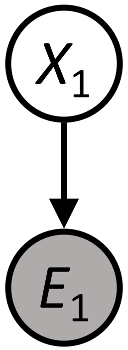
X: Tiles
E: 4 sensor readings

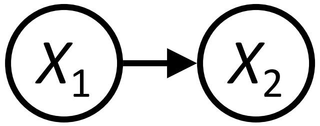




t=5

Inference: Base Cases





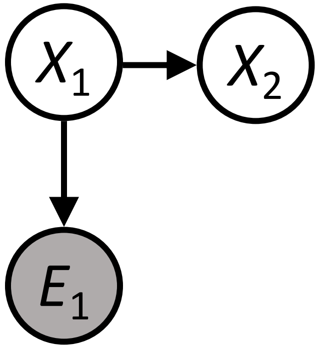
$$P(X_1|e_1)$$

$$\begin{aligned} P(x_1|e_1) &= P(x_1,e_1)/P(e_1) \\ &\propto_{X_1} P(x_1,e_1) \\ &= P(x_1)P(e_1|x_1) \end{aligned}$$

$$P(X_2)$$

$$\begin{aligned} P(x_2) &= \sum_{x_1} P(x_1,x_2) \\ &= \sum_{x_1} P(x_1)P(x_2|x_1) \end{aligned}$$

Inference: 1-Step State Evolution (Passage of Time)



$$P(x_1|e_1) = P(x_1, e_1)/P(e_1)$$
$$\propto_{X_1} P(x_1, e_1)$$
$$= P(x_1)P(e_1|x_1)$$

(marginalize) (chain rule)

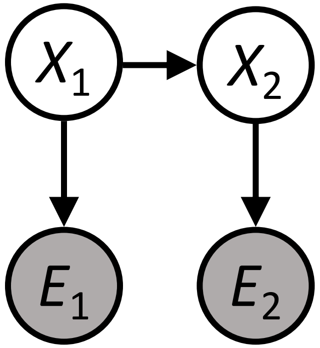
$$\underbrace{P(X_2|E_1 = e_1)}_{B'_2(X)} = \sum_{x_1} P(X_2, X_1 = x_1|E_1 = e_1) = \sum_{x_1} (P(X_2|X_1 = x_1, E_1 = e_1)P(X_1 = x_1|E_1 = e_1))$$

$$= \sum_{x_1} P(X_2|X_1 = x_1)\underbrace{P(X_1 = x_1|E_1 = e_1)}_{B_1(X)} = \alpha \sum_{x_1} P(X_2|X_1 = x_1)P(E_1 = e_1|X_1 = x_1)P(X_1 = x_1)$$

(markov property) (Bayes Rule + $P(E_1 = e_1)$ being constant)

$B_t(X) = P_t(X_t|e_1, \dots, e_t)$
$B'_t(X) = P_t(X_t|e_1, \dots, e_{t-1})$

Inference: 1-Step Observation



$$P(X_2|E_1 = e_1) = B_2'(X) = \sum_{x_1} P(X_2|X_1 = x_1)B_1(X)$$

(conditional probability) (evidences being constant)

$$P(X_2|E_1 = e_1, E_2 = e_2) = P(X_2, E_2 = e_2|E_1 = e_1)/P(E_2 = e_2|E_1 = e_1) = \alpha P(X_2, E_2 = e_2|E_1 = e_1)$$

$$= \alpha P(E_2 = e_2|X_2, E_1 = e_1)P(X_2|E_1 = e_1) = \alpha P(E_2 = e_2|X_2)P(X_2|E_1 = e_1) = \alpha P(E_2 = e_2|X_2)B_2'(X)$$

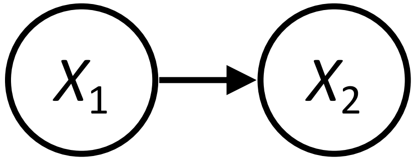
(chain rule) (conditional independence of emissions given state)

$B_t(X) = P_t(X_t|e_1, \ldots, e_t)$
$B'_t(X) = P_t(X_t|e_1, \ldots, e_{t-1})$

Passage of Time

- Assume we have current belief P(X | evidence to date)

$$B(X_t) = P(X_t|e_{1:t})$$



- Then, after one time step passes (no new evidence):

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

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

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

- Or compactly:

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$

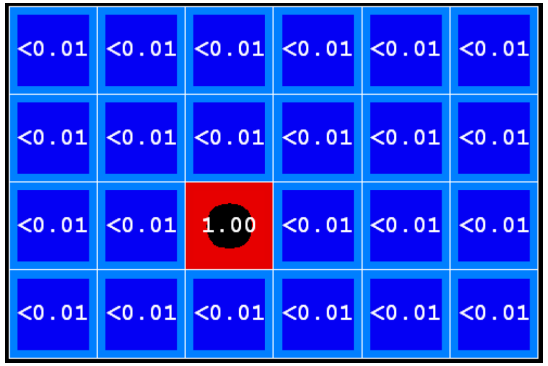
- Basic idea: beliefs get "pushed" through the transitions
  - With the "B" notation, we have to be careful about what time step t the belief is about, and what evidence it includes

Passage of Time

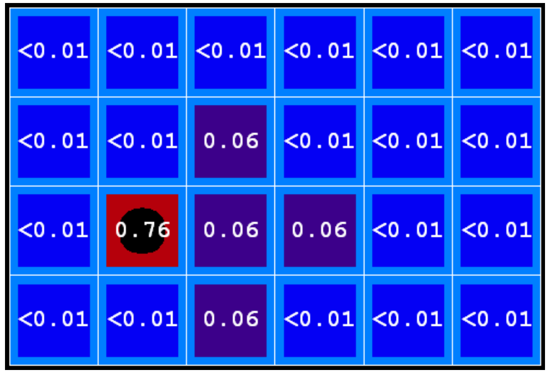
- What happens if we do not get any evidence moving forward?
- As time passes, uncertainty "accumulates"

$$B'(X_{t+1}) = \sum_{x_t} P(X'|x_t)B(x_t)$$

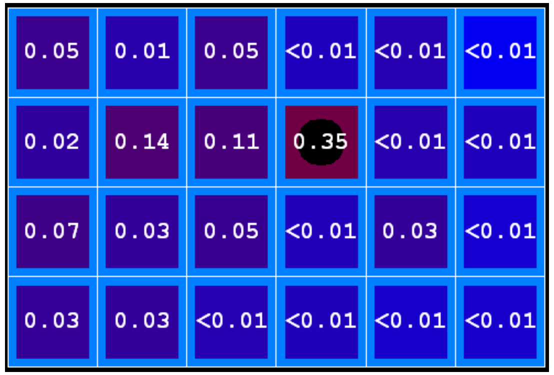
- E.g. a ghost that usually goes clockwise in Pacman, we have $B_1(X)$



T = 1



T = 2

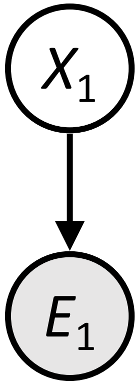


T = 5

Observation

- Assume we have current belief P(X | previous evidence):

$$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$



- Then, after evidence comes in:

$$\begin{aligned} P(X_{t+1}|e_{1:t+1}) = \ & P(X_{t+1}, e_{t+1}|e_{1:t})/P(e_{t+1}|e_{1:t}) \\ & \propto_{X_{t+1}} P(X_{t+1}, e_{t+1}|e_{1:t}) \\ & = P(e_{t+1}|e_{1:t}, X_{t+1})P(X_{t+1}|e_{1:t}) \\ & = P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t}) \end{aligned}$$

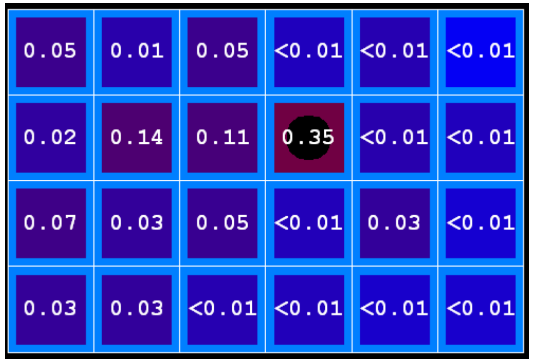
- Or, compactly:

$$B(X_{t+1}) \propto_{X_{t+1}} P(e_{t+1}|X_{t+1})B'(X_{t+1})$$

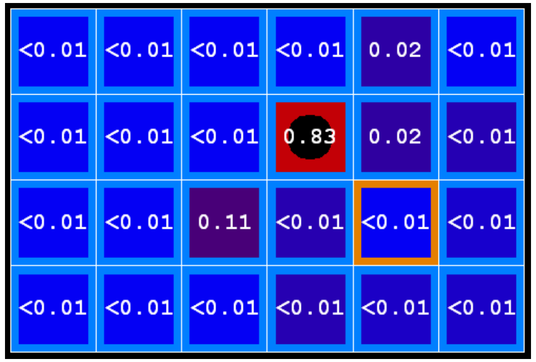
- Basic idea: beliefs "reweighted" by likelihood of evidence
- Unlike passage of time, we have to renormalize

Example Observation

- As we get observations, beliefs get reweighted, uncertainty “decreases”



Before observation



After observation

$$B(X) \propto P(e|X)B'(X)$$

The Forward Algorithm

- We are given evidence at each time and want to know

$$B_t(X) = P(X_t|e_{1:t})$$

- We can derive the following updates

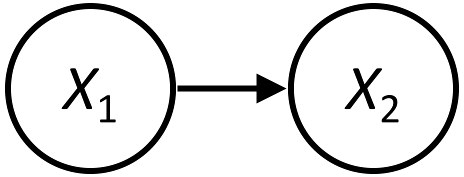
$$\begin{aligned}
P(x_t|e_{1:t}) &\propto_X P(x_t, e_{1:t}) \\
&= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t}) \\
&= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t) \\
&= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1})
\end{aligned}$$

We can normalize as we go if we want to have P(x|e) at each time step, or just once at the end…

Online Belief Updates

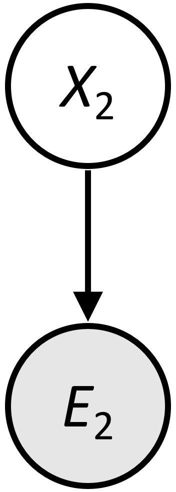
- Every time step, we start with current P(X | evidence)
- We update for time:

$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$



- We update for evidence:

$$P(x_t|e_{1:t}) \propto_X P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$



- The forward algorithm does both at once (and doesn't normalize)

Example: Weather HMM

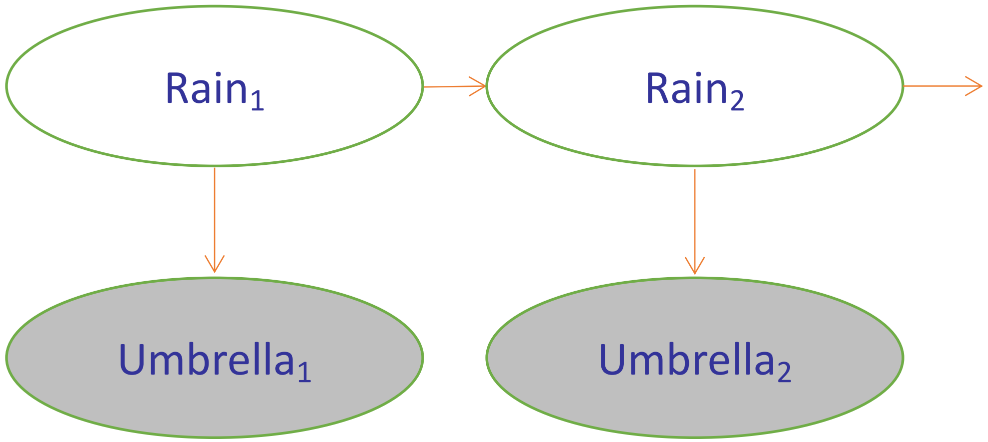
$P(+r) = 0.5$
$P(-r) = 0.5$

$B'(+r) = 0.5$
$B'(-r) = 0.5$

$B(+r) = 0.818$
$B(-r) = 0.182$

$B'(+r) = 0.627$
$B'(-r) = 0.373$

$B(+r) = 0.883$
$B(-r) = 0.117$



R_t	R_{t+1}	$P(R_{t+1} \mid R_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

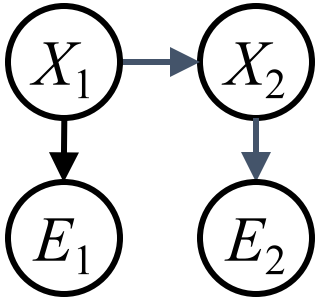
Filtering Example

Elapse time: compute P(X_t | $e_{1:t-1}$)

$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$

Observe: compute P(X_t | $e_{1:t}$)

$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$



		Belief: <P(rain), P(sun)>
Prior on X_1	$P(X_1)$	<0.5, 0.5>
Observe	$P(X_1 \mid E_1 = umbrella)$	<.9 .2><.5 .5> = <0.82, 0.18>
Elapse time	$P(X_2 \mid E_1 = umbrella)$	<.7 .3>*.82 + <.3 .7 >*.18 = <0.63, 0.37>
Observe	$P(X_2 \mid E_1 = umb, E_2 = umb)$	<.9 .2><.63 .37> = <0.88, 0.12>

Applied Example

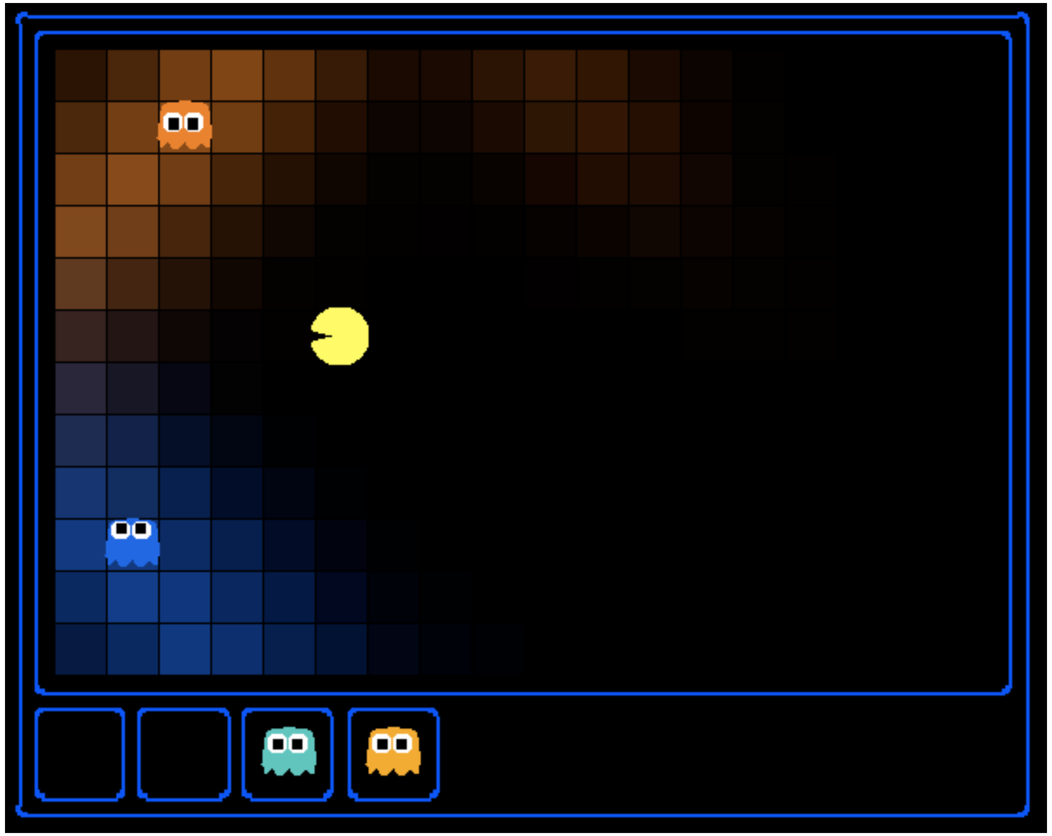
- Let’s have another go at the Markov Chains and the Hidden Markov Models

- The example will be about ghost tracking
- Markov Model: Probability of ghost location given ghost dynamics
- Hidden Markov Model: Tracking ghosts with sonar

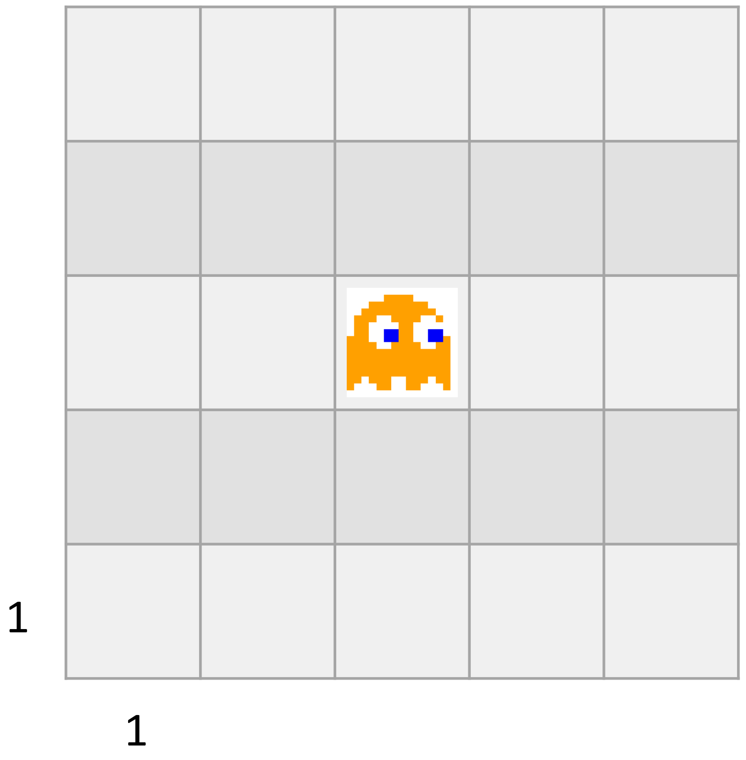
Myth of the Grandpac

- It is said that Pacman's grandfather, GrandPac learned to hunt ghosts without the use of special capsules
- With his power he started to hunt ghosts for sport
- He was blinded by his power and was only able to track ghosts by their sound (using his ears as sonar)

- It is also said that GrandPac was very smart and knew about probabilistic methods
- I will teach you the ways of the GrandPac…



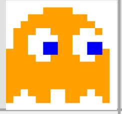
Markov Chain

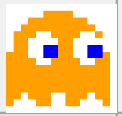


- States: The locations on the grid, e.g. X = (1,1), where the ghost can be
- Transition Model: Ghost dynamics i.e. how the ghost moves between states
- Initial State: We will assume this is given

Ghost Dynamics – Transition Model

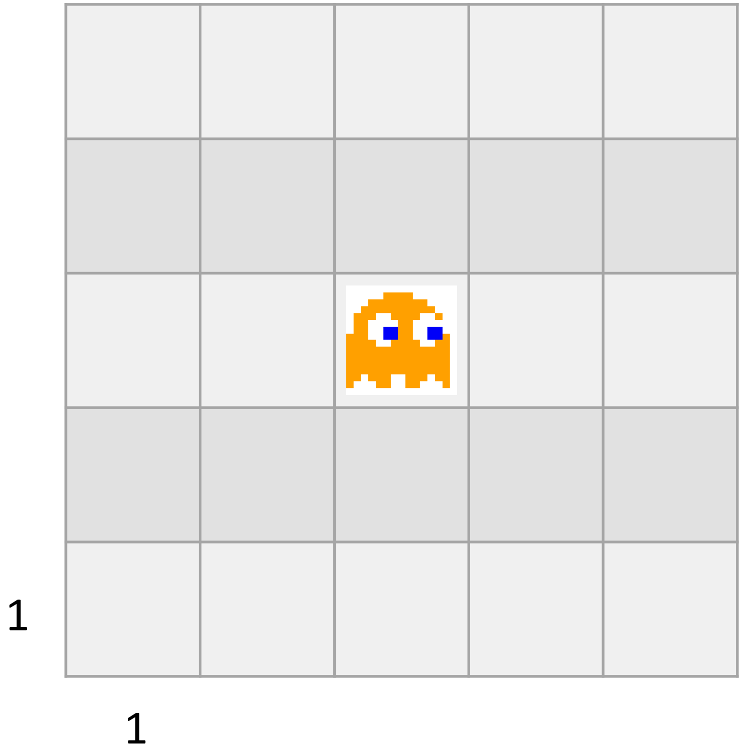
- Random ghosts: Move in any direction or stop randomly:

0	1/5	0
1/5	1/5 	1/5
0	1/5	0

1/3 	1/3
1/3	0

- Not shown but you can guess what the transition model would be if there was a wall

Where will the ghost be next?



T = 1

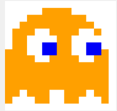
- Given $P\big(X_1 = (3,3)\big) = 1.0$
- And the ghost dynamics

0	1/5	0
1/5	1/5 	1/5
0	1/5	0

$$P\big(X_2 = (3,3)\big) = P\big(X_2 = (3,3) | X_1 = (3,3)\big) P\big(X_1 = (3,3)\big) = 0.2 \cdot 1.0$$
$$P\big(X_2 = (2,3)\big) = P\big(X_2 = (2,3) | X_1 = (3,3)\big) P\big(X_1 = (3,3)\big) = 0.2 \cdot 1.0$$

...

Where will the ghost be next?

T = 1

		0.2		
	0.2	0.2	0.2	
		0.2		

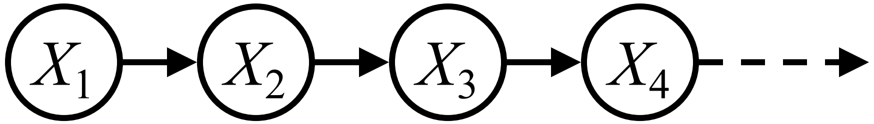
T = 2

?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?
?	?	?	?	?

T = 3

Recall: Mini-Forward Algorithm

- Question: What's P(X) on some t?



$$P(x_1) = \text{known}$$

$$P(x_t) = \sum_{x_{t-1}} P(x_{t-1}, x_t)$$

$$= \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1})$$

Forward simulation

Let's See T=3

$P(X_1 = (3,3)) = 1.0$

$P\left(X_2 = (3,2)\right) = P\left(X_2 = (2,3)\right) = P\left(X_2 = (4,3)\right) = P\left(X_2 = (3,4)\right) = 0.2$

$P(X_3 = (3,3)) = ?$

$$\begin{aligned} P(X_3 = (3,3)) &= P\left(X_3 = (3,3) \middle| X_2 = (3,3)\right) P\left(X_2 = (3,3)\right) \\ &+ P\left(X_3 = (3,3) \middle| X_2 = (3,2)\right) P\left(X_2 = (3,2)\right) \\ &+ P\left(X_3 = (3,3) \middle| X_2 = (2,3)\right) P\left(X_2 = (2,3)\right) \\ &+ P\left(X_3 = (3,3) \middle| X_2 = (4,3)\right) P\left(X_2 = (4,3)\right) \\ &+ P\left(X_3 = (3,3) \middle| X_2 = (3,4)\right) P\left(X_2 = (3,4)\right) \end{aligned}$$

$P(x_1) = $ known

$$P(x_t) = \sum_{x_{t-1}} P(x_{t-1}, x_t) = \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1})$$

For this problem, we only need to write the neighbors since other transitions have 0 probability

Let's See T=3

$$
\begin{aligned}
P(X_3 = (3,3)) &= P(X_3 = (3,3) \mid X_2 = (3,3))P(X_2 = (3,3)) \\
&+ P(X_3 = (3,3) \mid X_2 = (3,2))P(X_2 = (3,2)) \\
&+ P(X_3 = (3,3) \mid X_2 = (2,3))P(X_2 = (2,3)) \\
&+ P(X_3 = (3,3) \mid X_2 = (4,3))P(X_2 = (4,3)) \\
&+ P(X_3 = (3,3) \mid X_2 = (3,4))P(X_2 = (3,4)) \\
&= 0.2 \cdot 0.2 + 0.2 \cdot 0.2 + 0.2 \cdot 0.2 + 0.2 \cdot 0.2 + 0.2 \cdot 0.2 \\
&= 0.2
\end{aligned}
$$

Let’s See T=3

$$P(X_3 = (3,5)) = ?$$

$$\begin{aligned}P(X_3 = (3,5)) &= P\big(X_3 = (3,5) \big| X_2 = (3,5)\big)P\big(X_2 = (3,5)\big)\\ &+ P\big(X_3 = (3,5) \big| X_2 = (2,5)\big)P\big(X_2 = (2,5)\big)\\ &+ P\big(X_3 = (3,5) \big| X_2 = (3,4)\big)P\big(X_2 = (3,4)\big)\\ &+ P\big(X_3 = (3,5) \big| X_2 = (4,5)\big)P\big(X_2 = (4,5)\big)\\ &= 0.25 \cdot 0 + 0.25 \cdot 0 + 0.2 \cdot 0.2 + 0.25 \cdot 0\\ &= 0.04\end{aligned}$$

T = 2

		0.2		
	0.2	0.2	0.2	
		0.2		

$P(x_1) =$ known

$$P(x_t) = \sum_{x_{t-1}} P(x_{t-1}, x_t)$$
$$= \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1})$$

Note that transition from (3,4) to (3,5), not vice versa!

1/4	1/4 	1/4
0	1/4	0

Group Exercise

$P(X_3 = (3,4)) = ?$

$$
\begin{aligned}
P(X_3 = (3,4)) &= P\big(X_3 = (3,4) \big| X_2 = (3,4)\big) P\big(X_2 = (3,4)\big) \\
&+ P\big(X_3 = (3,4) \big| X_2 = (3,3)\big) P\big(X_2 = (3,3)\big) \\
&+ P\big(X_3 = (3,4) \big| X_2 = (4,4)\big) P\big(X_2 = (4,4)\big) \\
&+ P\big(X_3 = (3,4) \big| X_2 = (3,5)\big) P\big(X_2 = (3,5)\big) \\
&+ P\big(X_3 = (3,4) \big| X_2 = (2,4)\big) P\big(X_2 = (2,4)\big) \\
&= 0.2 \cdot 0.2 + 0.2 \cdot 0.2 + 0.2 \cdot 0.0 + 0.25 \cdot 0.0 + 0.2 \cdot 0.0 \\
&= 0.08
\end{aligned}
$$

T = 2

		0.2		
	0.2	0.2	0.2	
		0.2		

$P(x_1) =$ known

$$
\begin{aligned}
P(x_t) &= \sum_{X_{t-1}} P(x_{t-1}, x_t) \\
&= \sum_{X_{t-1}} P(x_t | x_{t-1}) P(x_{t-1})
\end{aligned}
$$

Home Exercise

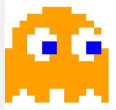
T = 2

		0.2		
	0.2	0.2	0.2	
		0.2		

$$P(X_3 = (2,4)) = ?$$

$$\begin{aligned} P(X_3 = (2,4)) &= P\big(X_3 = (2,4) \big| X_2 = (2,4)\big) P\big(X_2 = (2,4)\big) \\ &+ P\big(X_3 = (2,4) \big| X_2 = (3,4)\big) P\big(X_2 = (3,4)\big) \\ &+ P\big(X_3 = (2,4) \big| X_2 = (2,5)\big) P\big(X_2 = (2,5)\big) \\ &+ P\big(X_3 = (2,4) \big| X_2 = (1,4)\big) P\big(X_2 = (1,4)\big) \\ &+ P\big(X_3 = (2,4) \big| X_2 = (2,3)\big) P\big(X_2 = (2,3)\big) \\ &= 0.2 \cdot 0.0 + 0.2 \cdot 0.2 + 0.25 \cdot 0.0 + 0.25 \cdot 0.0 + 0.2 \cdot 0.2 \\ &= 0.08 \end{aligned}$$

Where will the ghost be next?

T = 1

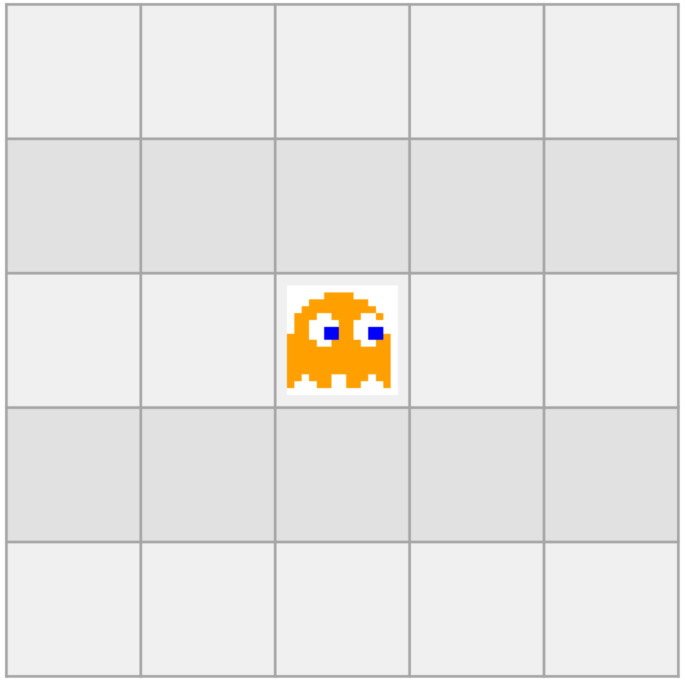
		0.2		
	0.2	0.2	0.2	
		0.2		

T = 2

0	0	0.04	0	0
0	0.08	0.08	0.08	0
0.04	0.08	0.2	0.08	0.04
0	0.08	0.08	0.08	0
0	0	0.04	0	0

T = 3

Where will the ghost be next?



T = 1

0	0.03	0.03	0.03	0
0.03	0.05	0.1	0.05	0.03
0.03	0.1	0.1	0.1	0.03
0.03	0.05	0.1	0.05	0.03
0	0.03	0.03	0.03	0

T = 4

0.01	0.02	0.04	0.02	0.01
0.02	0.06	0.07	0.06	0.02
0.04	0.07	0.1	0.07	0.04
0.02	0.06	0.07	0.06	0.02
0.01	0.02	0.04	0.02	0.01

T = 5

Uncertainty accumulates
Bonus points: What is the stationary distribution? (Hint: It is not uniform)

Let’s add the Sonar

It is clear that we will not get very far if we do not sense the ghost!

P(E|X)=

		3/64		
	1/16	3/32	1/16	
3/64	3/32	3/16	3/32	3/64
	1/16	3/32	1/16	
		3/64		

This table gives us the probability of getting a sonar reading at a given state if the ghost is at state (3,3)

We are going to assume that the boundaries absorb all the sound

We are not going to change the distribution if the ghost is adjacent to the boundaries (why would this work?)

The Forward Algorithm

- We are given evidence at each time and want to know

$$B_t(X) = P(X_t|e_{1:t}) \longrightarrow \text{Where is the ghost?}$$

- We can derive the following updates

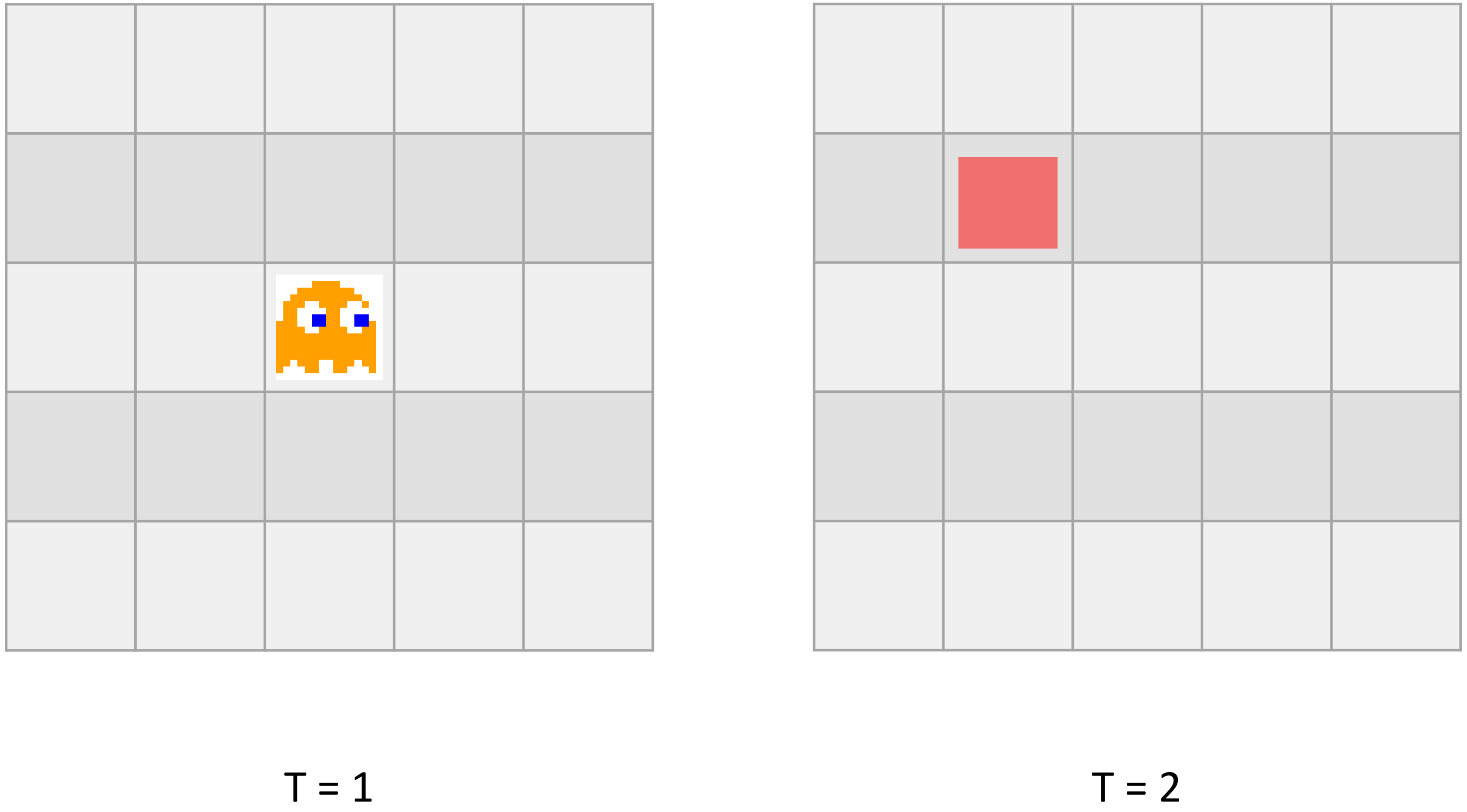
$$P(x_t|e_{1:t}) \propto_X P(x_t, e_{1:t})$$

$$= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t})$$

$$= \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) P(x_t|x_{t-1}) P(e_t|x_t)$$

$$= \underbrace{P(e_t|x_t)}_{\text{Sensor Observation}} \underbrace{\sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1})}_{\text{Ghost Dynamics Passage of Time}}$$

Where will the ghost be next?



What is the distribution now that we got a sonar reading at (2,4)?

Where is this ghost?!

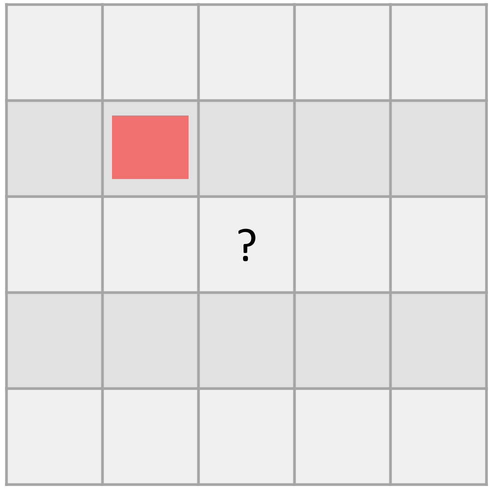
$$P(x_t|e_{1:t}) \propto_X P(x_t, e_{1:t})$$
$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1})$$

$P(X_1 = (3{,}3)) = 1.0$

$P(X_2 = (3{,}3)|e_{1:2}) = P(X_2 = (3{,}3)|E_2 = (2{,}4)) = ?$

(assume $P(E_1 = (3{,}3)|X_1 = (3{,}3)) = 1.0$)

$$\begin{aligned}
B(X_2 = (3{,}3)) \quad \propto\ & P(E_2 = (2{,}4)|X_2 = (3{,}3)) \\
& \text{x}\, [P(X_2 = (3{,}3)|X_1 = (3{,}3)) P(X_1 = (3{,}3)) \\
& + P(X_2 = (3{,}3)|X_1 = (3{,}2)) P(X_1 = (3{,}2)) \\
& + P(X_2 = (3{,}3)|X_1 = (2{,}3)) P(X_1 = (2{,}3)) \\
& + P(X_2 = (3{,}3)|X_1 = (4{,}3)) P(X_1 = (4{,}3)) \\
& + P(X_2 = (3{,}3)|X_1 = (3{,}4)) P(X_1 = (3{,}4))] \\
& = \frac{1}{16}\ \text{x}\ (0.2 \cdot 1.0 + 0.2 \cdot 0 + 0.2 \cdot 0 + 0.2 \cdot 0) \\
& = \frac{1}{16} \text{x}\ 0.2 = \frac{1}{80}
\end{aligned}$$



		3/64		
	1/16	3/32	1/16	
3/64	3/32	3/16	3/32	3/64
	1/16	3/32	1/16	
		3/64		

Sonar, not the grid!

Group Exercise

$$P(x_t|e_{1:t}) \propto_X P(x_t, e_{1:t})$$
$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}, e_{1:t-1})$$

$B(X_2 = (2,3)) = B(X_2 = (3,4)) = ?$

$B(X_2 = (2,3)) \propto \frac{3}{32} \times 0.2 = 3/160$

What about the rest?

0!

		?		
	?			

		3/160		
	3/160	2/160		

Normalize

		0.375		
	0.375	0.25		

		3/64		
	1/16	3/32	1/16	
3/64	3/32	3/16	3/32	3/64
	1/16	3/32	1/16	
		3/64		

Sonar, not the grid!

Home Exercise

$B(X_3|e_{1:3}) =?\ E_3 = (2,3)$

		0.375		
	0.375	0.25		

		3/64		
	1/16	3/32	1/16	
3/64	3/32	3/16	3/32	3/64
	1/16	3/32	1/16	
		3/64		

0	1/5	0
1/5	1/5	1/5
0	1/5	0

	0.1682	0.0935		
0.0841	0.2804	0.2243	0.028	
	0.0841	0.0374		

Approximate Filtering

- Filtering: $B_t(X) = \mathrm{P}(X_t|e_1, \ldots, e_t)$, what does this look like?
    - $\mathrm{P}(Q|E_1 = e_1, \ldots, E_k = e_k)$, i.e. inference

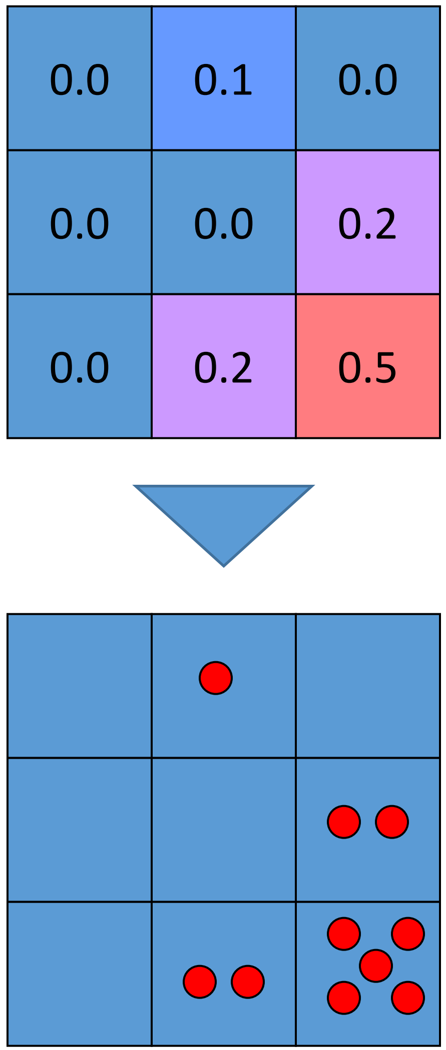
- Forward Algorithm: Exact Inference for Filtering

- The size of the state to track may be too big for exact inference

- A sample-based approach: Particle Filters

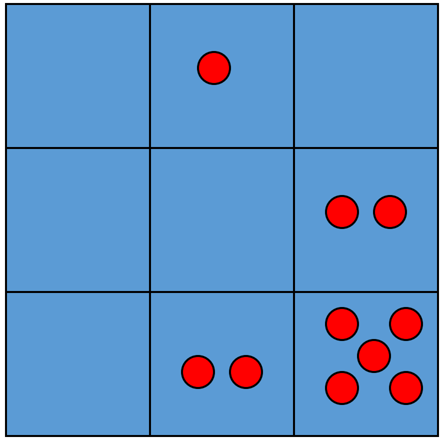
Particle Filtering

- Sometimes |X| is too big to use exact inference
  - |X| may be too big to even store B(X)
  - E.g. X is continuous
- Solution: approximate inference
  - Track samples of X, not all values
  - Samples are called particles
  - Time per step is linear in the number of samples
  - But, number needed may be large
  - In memory: list of particles, not states
- This is how robot localization works in practice
- Particle is just new name for sample, (we will see why)



Representation: Particles

- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|
  - Storing map from X to counts would defeat the point
- P(x) approximated by number of particles with value x
  - Many x will have P(x) = 0!
  - More particles, more accuracy
  - P(x=(3,3)) = 5/10
- For now, all particles have a weight of 1



10 Particles = **samples of X**:

(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(2,1)
(3,3)
(3,3)
(2,1)

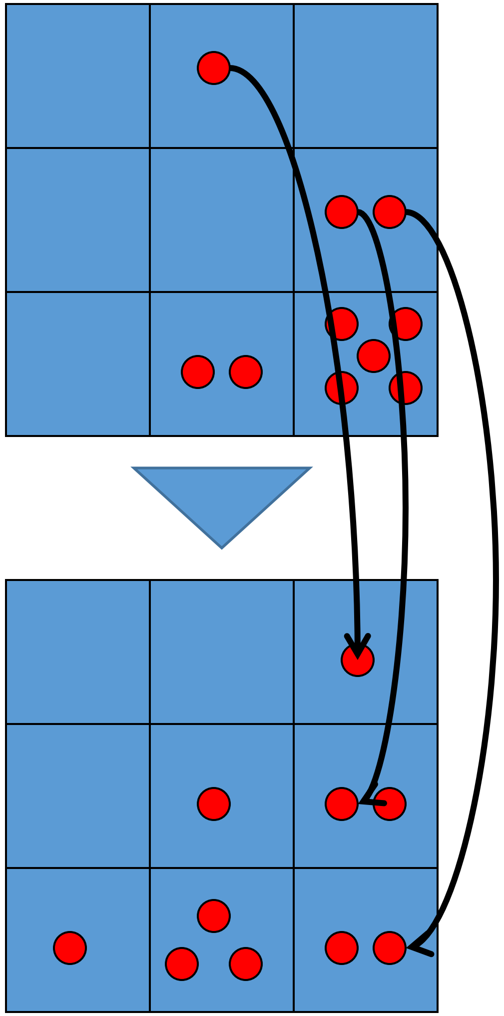
Particle Filtering: Elapse Time

- Each particle is moved by sampling its next position from the transition model

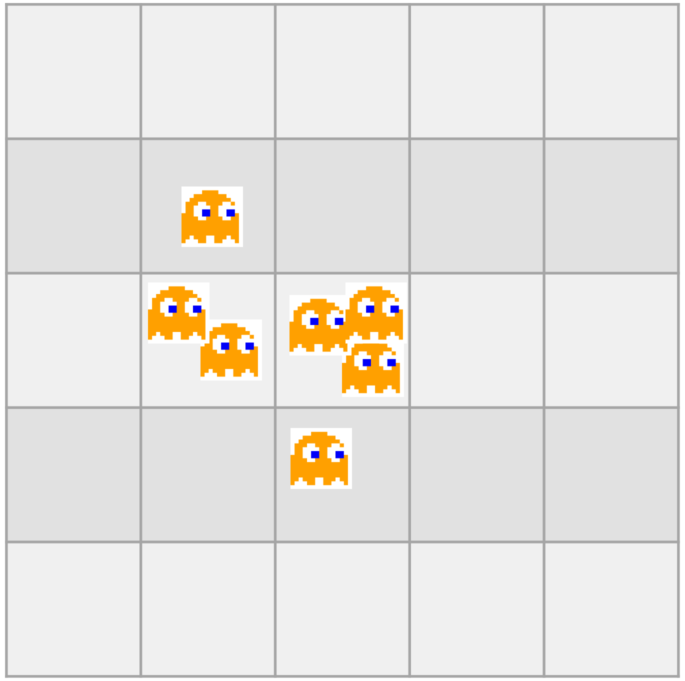
$$x' = \text{sample}(P(X'|x))$$

  - This is like prior sampling – samples' frequencies reflect the transition probabilities
  - Here, most samples move clockwise, but some move in another direction or stay in place

- This captures the passage of time
  - If we have enough samples, close to the exact values before and after (consistent)



Example: Ghost



If the ghosts represent particles, how would we sample their next states?

Ghost dynamics!

0	1/5	0
1/5	1/5 	1/5
0	1/5	0

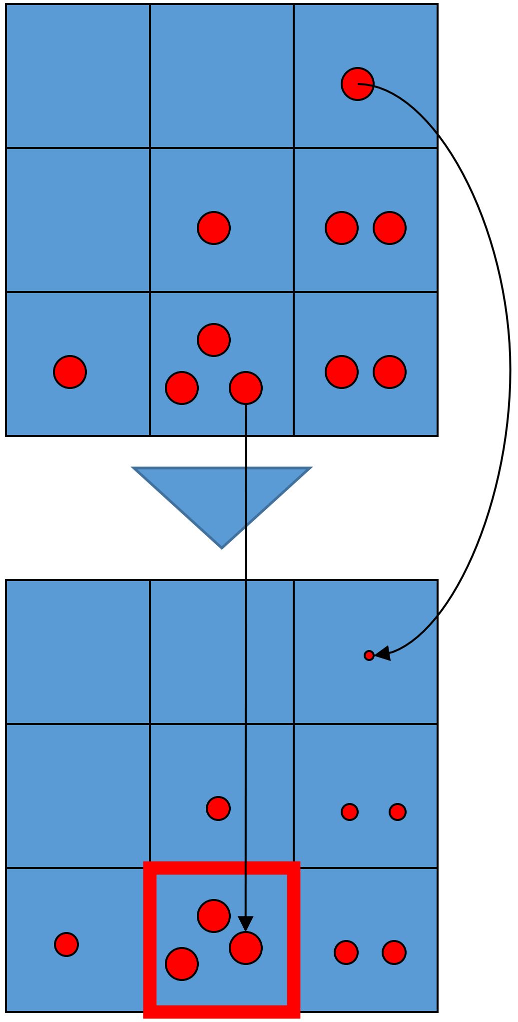
Particle Filtering: Observe

- Slightly trickier:
  - Don't sample the observation, it's fixed!
  - This is like likelihood weighting, we down-weight our samples based on the evidence

$$w(x) = P(e|x)$$

$$B(X) \propto P(e|X)B'(X)$$

  - As before, the probabilities don't sum to one, since most have been down-weighted (in fact they sum to an approximation of P(e))



Example: Sonar

We would use the sonar model to assign weights to samples:

		3/64		
	1/16	3/32	1/16	
3/64	3/32	3/16	3/32	3/64
	1/16	3/32	1/16	
		3/64		

A sensor model that gives you 0 distribution can be problematic!
With bad particle distribution, you might lose a lot of particles

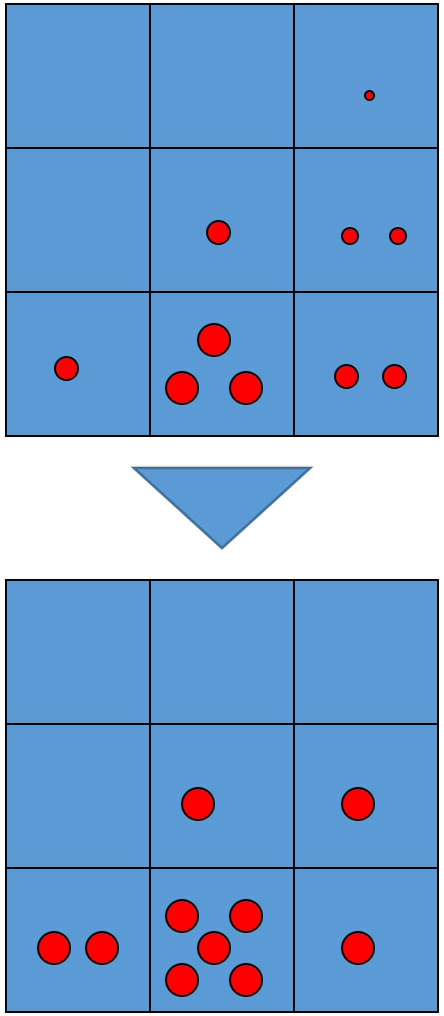
This problem is an instance of *sample impoverishment*

Particle Filtering: Resample

- Rather than tracking weighted samples, we resample
- N times, we choose from our weighted sample distribution (i.e., draw with replacement)
- This is equivalent to renormalizing the distribution
- Now the update is complete for this time step, continue with the next one

Old Particles:
(3,3) w=0.1
(2,1) w=0.9
(2,1) w=0.9
(3,1) w=0.4
(3,2) w=0.3
(2,2) w=0.4
(1,1) w=0.4
(3,1) w=0.4
(2,1) w=0.9
(3,2) w=0.3

New Particles:
(2,1) w=1
(2,1) w=1
(2,1) w=1
(3,2) w=1
(2,2) w=1
(2,1) w=1
(1,1) w=1
(3,1) w=1
(2,1) w=1
(1,1) w=1



How to resample?

Remember the cumulative distribution and sampling?

1. Normalize the weights
   - Sum them up
   - Divide each weight by the sum
2. Calculate an array of the cumulative sum of the weights
3. Generate a uniform sample and the find which range of the array it falls into
4. The index of that range corresponds to the new particle
5. Repeat until you have the desired number of samples.

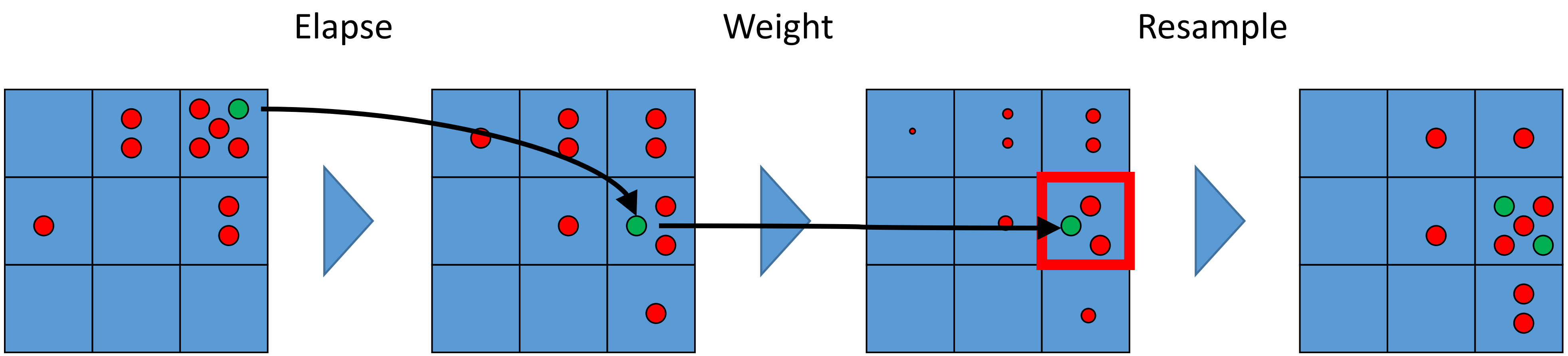
How to resample?

Particle Weights	Normalized Weights	Range	Uniform samples	New Particles:
(3,3) w=0.1	(3,3) w=0.02	(3,3) w=[0,0.02)	0.21	(2,1) w=1
(2,1) w=0.9	(2,1) w=0.18	(2,1) w=[0.02,0.2)	0.023	(2,1) w=1
(2,1) w=0.9	(2,1) w=0.18	(2,1) w=[0.2,0.38)	0.81	(2,1) w=1
(3,1) w=0.4	(3,1) w=0.08	(3,1) w=[0.38,0.46)	0.49	(3,2) w=1
(3,2) w=0.3	(3,2) w=0.06	(3,2) w=[0.46,0.52)	0.54	(2,2) w=1
(2,2) w=0.4	(2,2) w=0.08	(2,2) w=[0.52,0.6)	0.77	(2,1) w=1
(1,1) w=0.4	(1,1) w=0.08	(1,1) w=[0.6,0.68)	0.63	(1,1) w=1
(3,1) w=0.4	(3,1) w=0.08	(3,1) w=[0.68,0.76)	0.41	(3,1) w=1
(2,1) w=0.9	(2,1) w=0.18	(2,1) w=[0.76,0.94)	0.24	(2,1) w=1
(3,2) w=0.3	(3,2) w=0.06	(3,2) w=[0.94,1.0)	0.67	(1,1) w=1
Sum = 5			These are generated randomly. I am only giving examples	

We tend to lose sample diversity with this resampling. Another instance of sample impoverishment.
There are other particle filter variants to deal with this problem.

Particle Filtering Summary

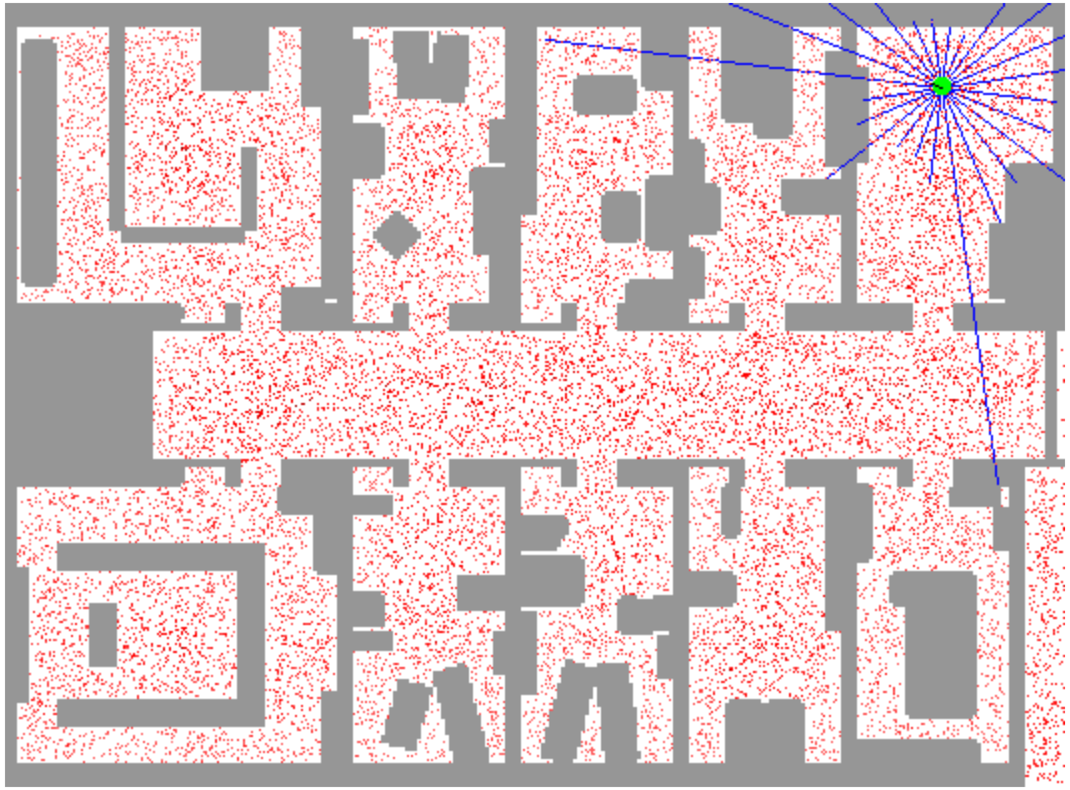
- Particles: track samples of states rather than an explicit distribution



Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,2)	(2,2) w=.4	(3,2)

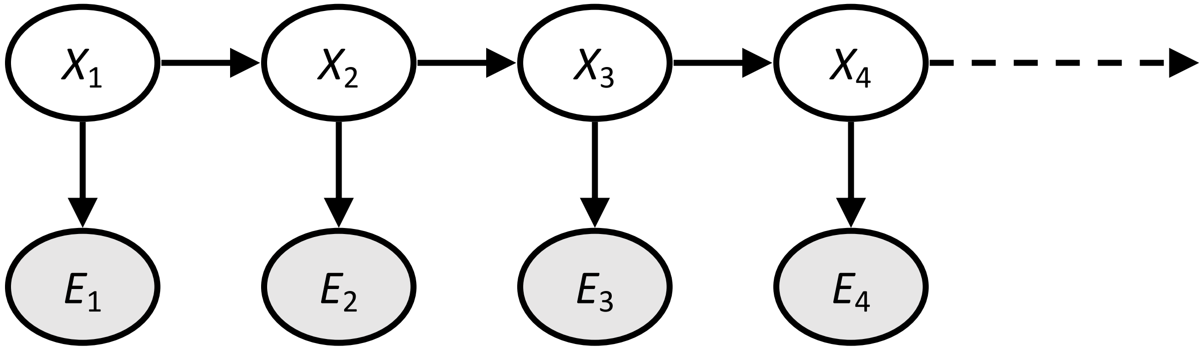
Robot Localization

- In robot localization:
  - We know the map, but not the robot's position
  - Observations may be vectors of range finder or sonar readings
  - Transition model is based on robot wheel encoders and/or kinematics
  - State space and readings are typically continuous (works basically like a very fine grid) so we cannot store B(X)
  - Particle filtering is a very popular technique for this!



HMMs: Most Likely Explanation Queries

- HMMs defined by
  - States X
  - Observations E
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X|X_{-1})$
  - Emissions: $P(E|X)$

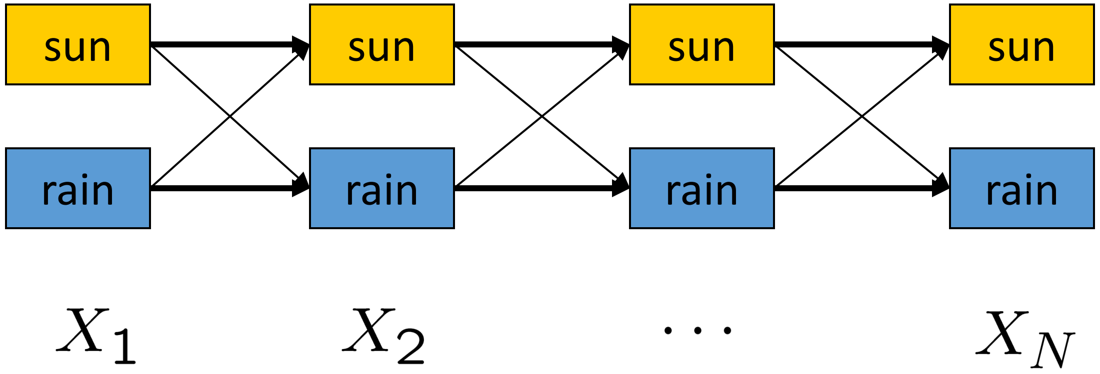


- New query: most likely explanation: $\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$

- New method: the Viterbi algorithm

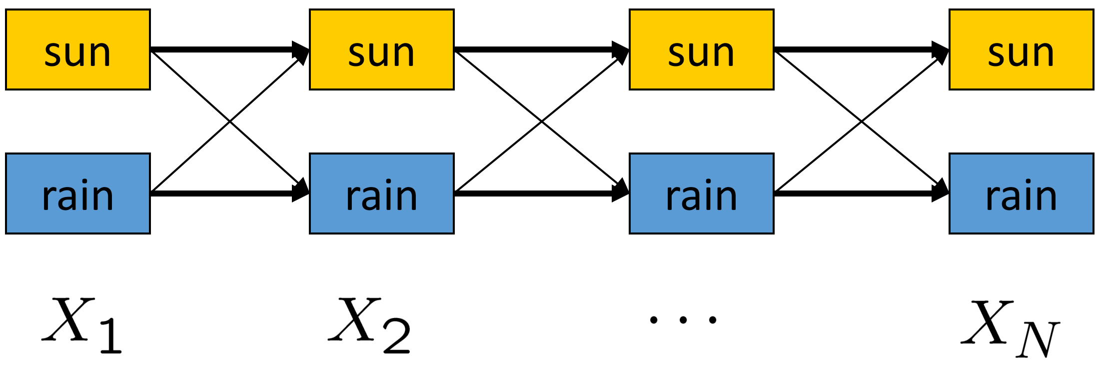
State Trellis

- State trellis: graph of states and transitions over time



- Each arc represents some transition $x_{t-1} \to x_t$
- Each arc has weight $P(x_t|x_{t-1})P(e_t|x_t)$
- Each path is a sequence of states
- The product of weights on a path is that sequence's probability along with the evidence
- Forward algorithm computes sums of paths, Viterbi computes best paths

Forward vs Viterbi



Forward Algorithm (Sum)

$$f_t[x_t] = P(x_t, e_{1:t})$$

$$= P(e_t|x_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) f_{t-1}[x_{t-1}]$$

Viterbi Algorithm (Max)

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

Viterbi Example

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$

$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

		1.0		

$e_2 = (2,4)$

		3/160		
	3/160	1/80		

$e_3 = (2,3)$

	0.00035	0.00023		
0.00035	0.0007	0.00035	0.00012	
	0.00035	0.00016		

$$m_1((1,1)) = 1.0$$

		3/64		
	1/16	3/32	1/16	
3/64	3/32	3/16	3/32	3/64
	1/16	3/32	1/16	
		3/64		

$$m_2((1,1)) = \frac{1}{16} 0.2$$

$$m_2((2,3)) = \frac{3}{32} 0.2$$

$$m_2((3,4)) = \frac{3}{32} 0.2$$

$$m_3((2,3)) = \frac{3}{16} \max\left(0.2 \frac{1}{80}, 0.2 \frac{3}{160}, 0 \frac{3}{160}\right)$$

...

You should also keep track of the argmax then move backwards from the state with the highest m-value to get the path!

What to do with HMMs?

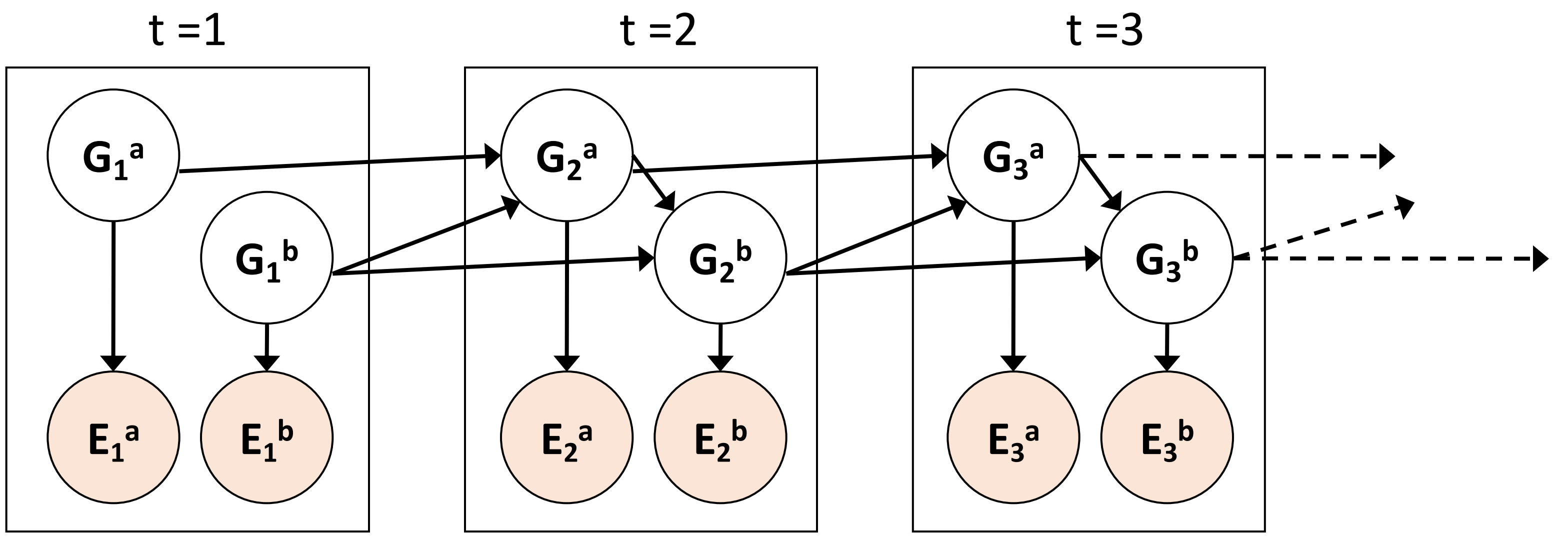
- Probability of an observed sequence $P(e_{1:t}) = \sum_X P(e_{1:t}|X)P(X)$
  - Using the forward algorithm
- Filtering $P(x_t|e_{1:t})$
  - Using the forward algorithm
- Smoothing $P(x_k|e_{1:t}), k < t$
  - Using something called the forward-backward algorithm
- Most likely explanation $\underset{x_{1:t}}{\operatorname{argmax}}(P(x_{1:t}|e_{1:t}))$
  - Using the Viterbi algorithm
- Generating Samples:
  - Apply prior sampling on the BN!

Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence

- Idea: Repeat a fixed Bayes net structure at each time

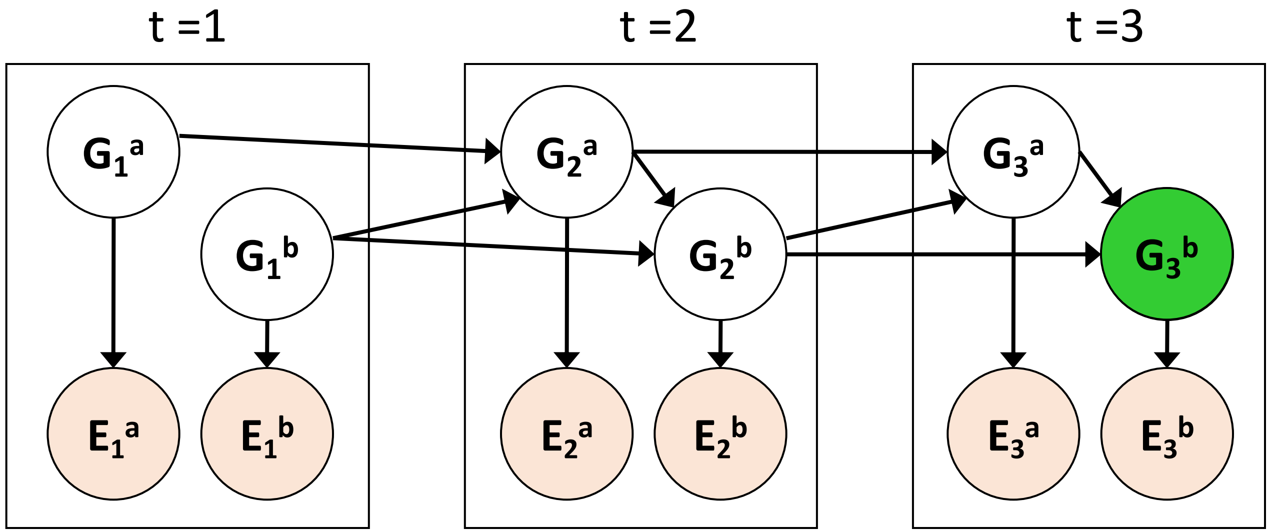
- Variables from time *t* can condition on those from *t-1*



- Dynamic Bayes nets are a generalization of HMMs

Exact Inference in DBNs

- Variable elimination applies to dynamic Bayes nets
- Procedure: "unroll" the network for T time steps, then eliminate variables until $P(X_T | e_{1:T})$ is computed



- Online belief updates: Eliminate all variables from the previous time step; store factors for current time only

DBN Particle Filters

- A particle is a complete sample for a time step

- **Initialize**: Generate prior samples for the t=1 Bayes net
  - Example particle: $\mathbf{G_1}^a = (3,3)$ $\mathbf{G_1}^b = (5,3)$

- **Elapse time**: Sample a successor for each particle
  - Example successor: $\mathbf{G_2}^a = (2,3)$ $\mathbf{G_2}^b = (6,3)$

- **Observe**: Weight each <u>*entire*</u> sample by the likelihood of the evidence conditioned on the sample
  - Likelihood: $P(\mathbf{E_1}^a \mid \mathbf{G_1}^a) * P(\mathbf{E_1}^b \mid \mathbf{G_1}^b)$

- **Resample:** Select prior samples (tuples of values) in proportion to their likelihood