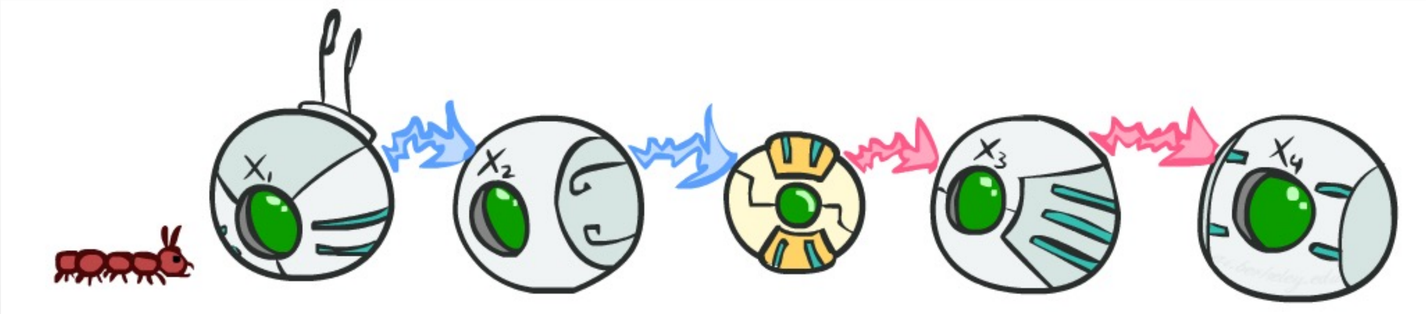
Artificial Intelligence
CE-417, Group 1
Computer Eng. Department
Sharif University of Technology

Spring 2024

By Mohammad Hossein Rohban, Ph.D.

Courtesy: Most slides are adopted from CSE-573 (Washington U.), original slides for the textbook, and CS-188 (UC. Berkeley).

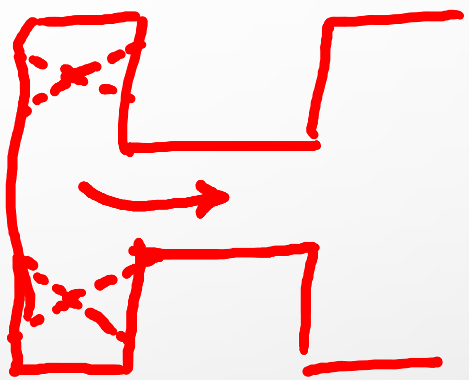
Temporal Probability Models



Markov Models

Reasoning over Time or Space

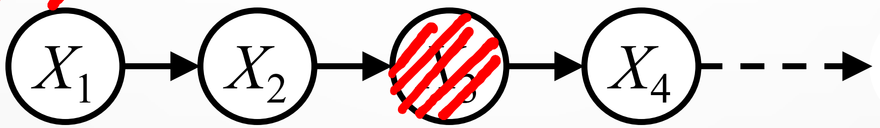
- Often, we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - User attention
  - Medical monitoring



- Need to introduce time (or space) into our models

Markov Models

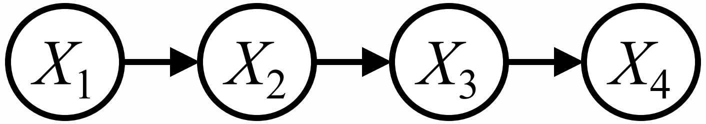
- Value of X at a given time is called the state



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

- Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial state probabilities)
- Stationarity assumption: transition probabilities the same at all times

Joint Distribution of a Markov Model



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

- Joint distribution:

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

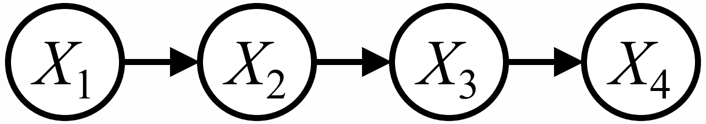
- More generally:

$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2|X_1)P(X_3|X_2)\ldots P(X_T|X_{T-1})$$

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

- Questions to be resolved:
  - Does this indeed define a joint distribution?
  - Can every joint distribution be factored this way, or are we making some assumptions about the joint distribution by using this factorization?

Chain Rule and Markov Models



- From the chain rule, every joint distribution over X_1, X_2, X_3, X_4 can be written as:

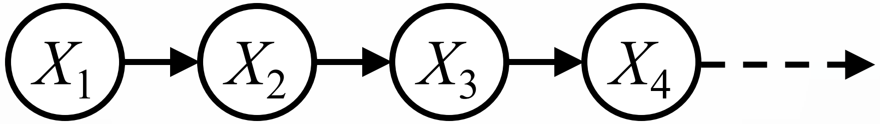
$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)P(X_4|X_1, X_2, X_3)$$

- Assuming that $X_3 \perp\!\!\!\perp X_1 \mid X_2$ and $X_4 \perp\!\!\!\perp X_1, X_2 \mid X_3$

Results in the expression posited on the previous slide:

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

Chain Rule and Markov Models



- From the chain rule, every joint distribution over $X_1, X_2, \ldots, X_T$ can be written as:

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

- Assuming that for all *t*:

$$X_t \perp\!\!\!\perp X_1, \ldots, X_{t-2} \mid X_{t-1}$$

Gives us the expression posited on the earlier slide:

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

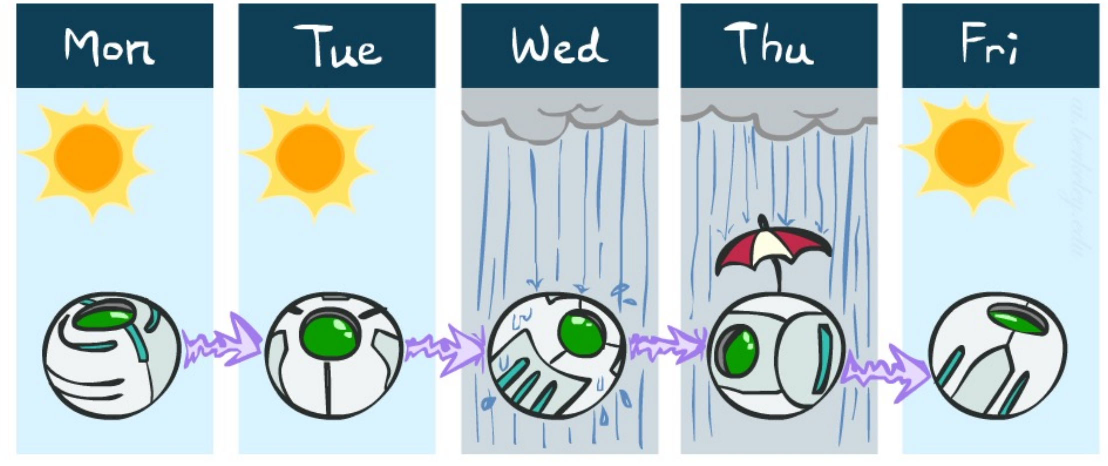
Example Markov Chain: Weather

- States: X = {rain, sun}

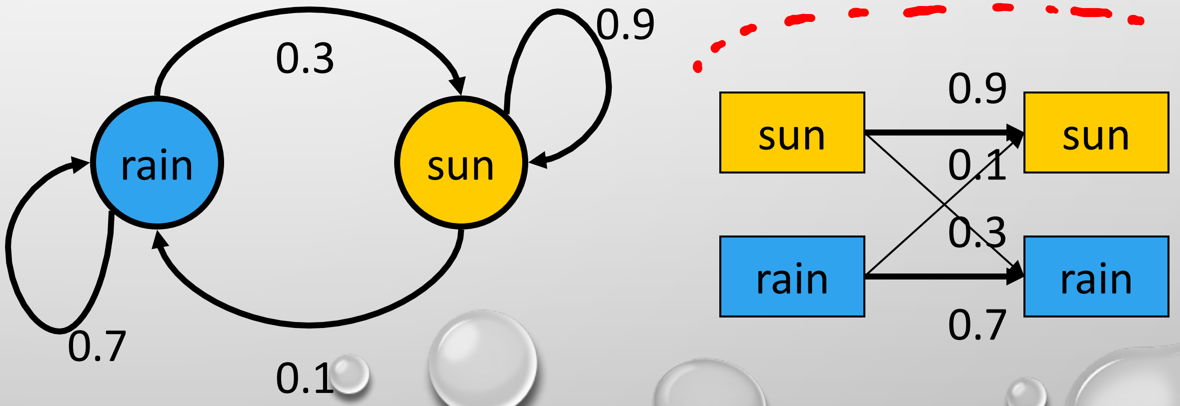
- Initial distribution: 1.0 sun

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

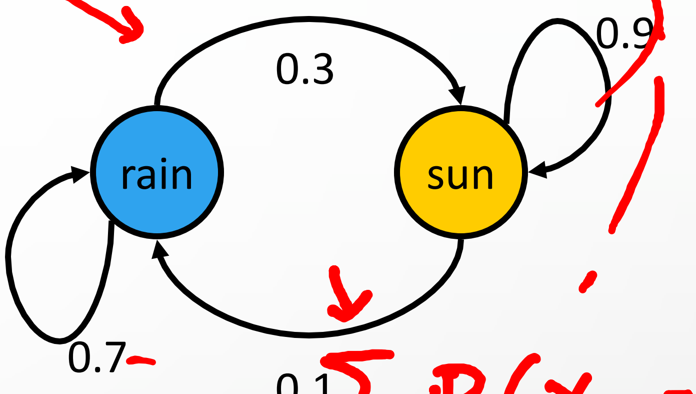
X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7



Two new ways of representing the same CPT



Example Markov Chain: Weather



- Initial distribution: 1.0 sun

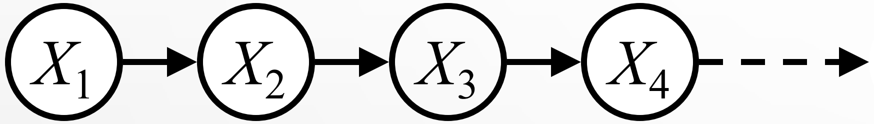
- What is the probability distribution after one step?

$$P(X_2 = \text{sun}) = P(X_2 = \text{sun} | X_1 = \text{sun}) P(X_1 = \text{sun}) + P(X_2 = \text{sun} | X_1 = \text{rain}) P(X_1 = \text{rain})$$

$$0.9 \cdot 1.0 + 0.3 \cdot 0.0 = 0.9$$

Mini-Forward Algorithm

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

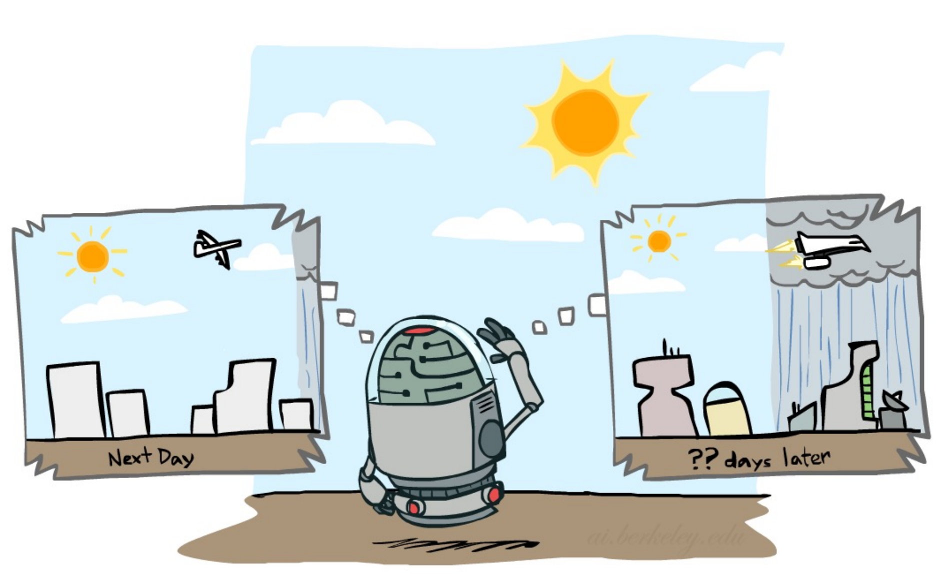


$P(x_1) = \text{known}$

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

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

Forward simulation



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 $P(X_1)$:

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

Stationary Distributions

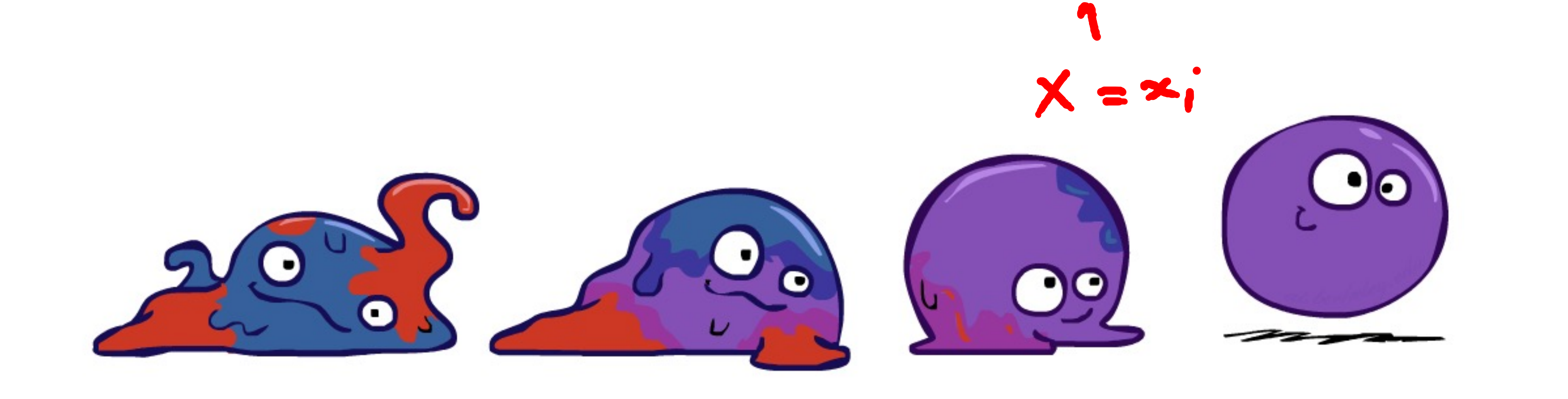
- For most chains:
  - Influence of the initial distribution gets less and less over time.
  - The distribution we end up in is independent of the initial distribution

- Stationary distribution:
  - The distribution we end up with is called the stationary distribution P_∞ of the chain
  - It satisfies $P_{t+1}(x) = \sum P(x|x) P_t(x)$

n eqs.

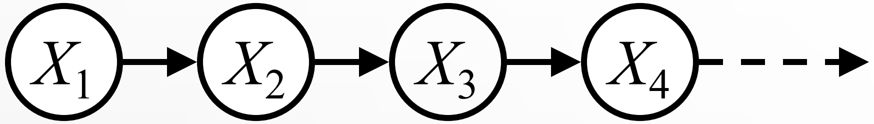
$$P_\infty(X) = P_{\infty+1}(X) = \sum_x P(X|x) P_\infty(x)$$

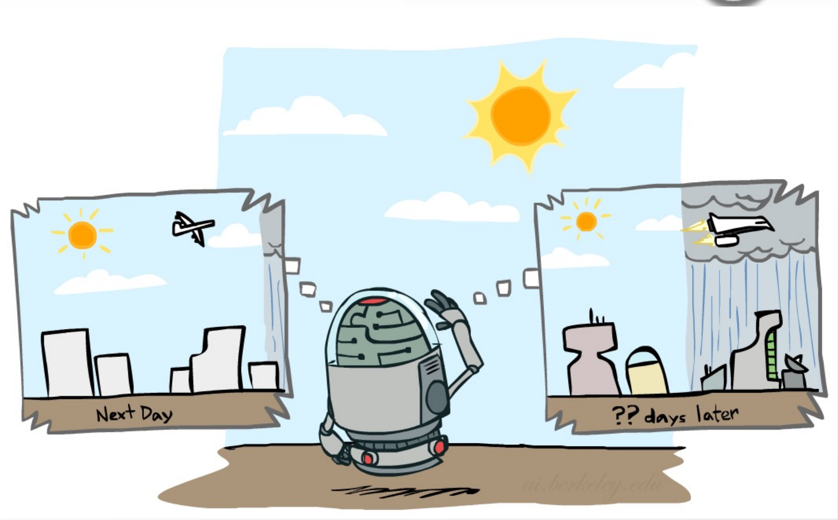
$X = x_i$



Example: Stationary Distributions

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





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

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

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

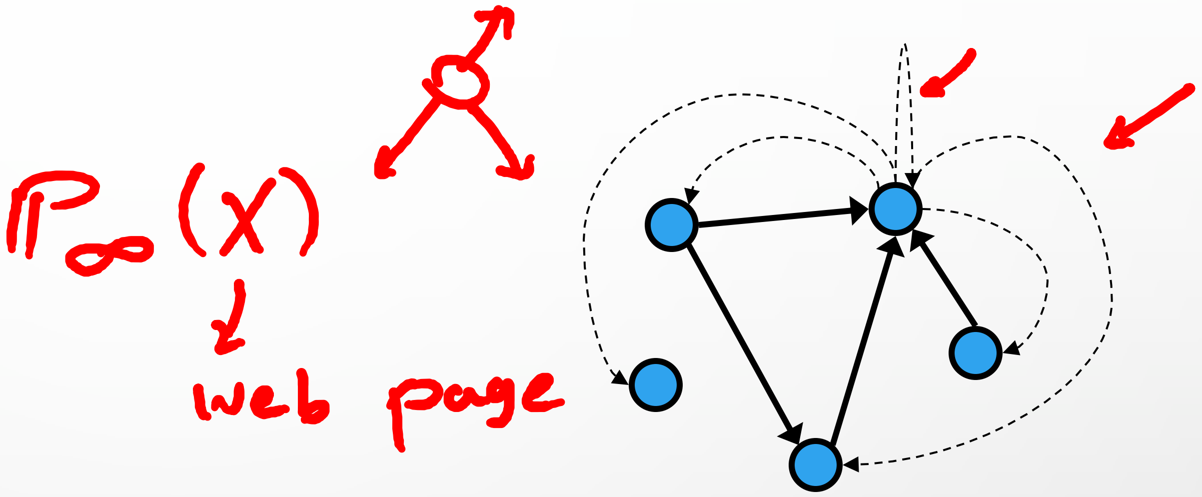
Also: $P_\infty(sun) + P_\infty(rain) = 1$

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

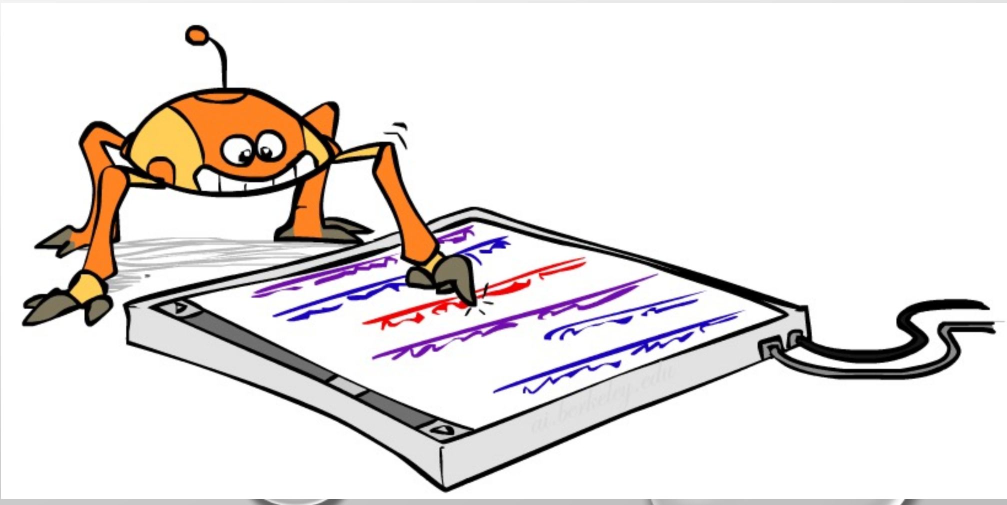
X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7

Application of Stationary Distribution: Web Link Analysis

- PageRank over a web graph
  - Each web page is a state
  - Initial distribution: uniform over pages
  - Transitions:
    - With prob. c, uniform jump to a random page (dotted lines, not all shown)
    - With prob. 1-c, follow a random outlink (solid lines)



- Stationary distribution
  - Will spend more time on highly reachable pages
  - e.g. Many ways to get to the acrobat reader download page
  - Somewhat robust to link spam
  - Google 1.0 returned the set of pages containing all your keywords in decreasing rank, now all search engines use link analysis along with many other factors (rank actually getting less important over time)



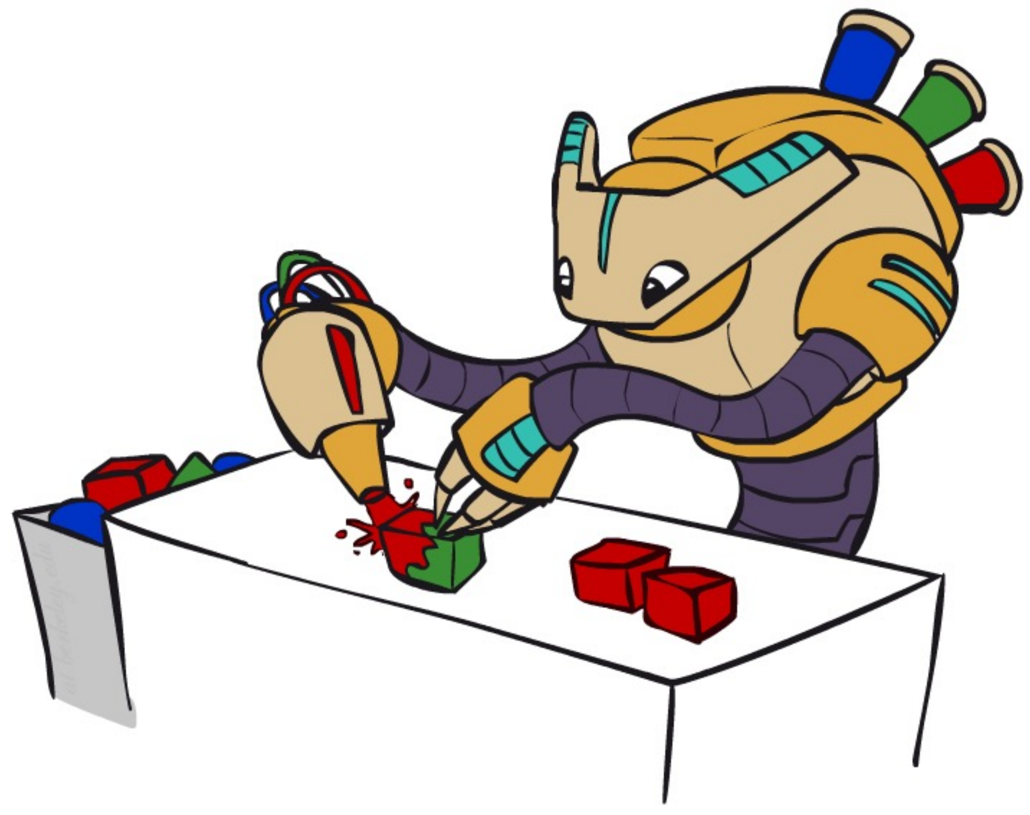
Application of Stationary Distributions: Gibbs Sampling

- Each joint instantiation over all hidden and query variables is a state: $\{X_1, \ldots, X_n\} = H \cup Q$

- Transitions:
  - Resample variable x_i according to

    $$p(X_i \mid X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n, E_1, \ldots, E_m)$$

- Stationary distribution:
  - Conditional distribution $P(X_1, X_2, \ldots, X_n | E_1, \ldots, E_m)$
  - Means that when running Gibbs sampling long enough we get a sample from the desired distribution
  - Requires some proof to show this is true!

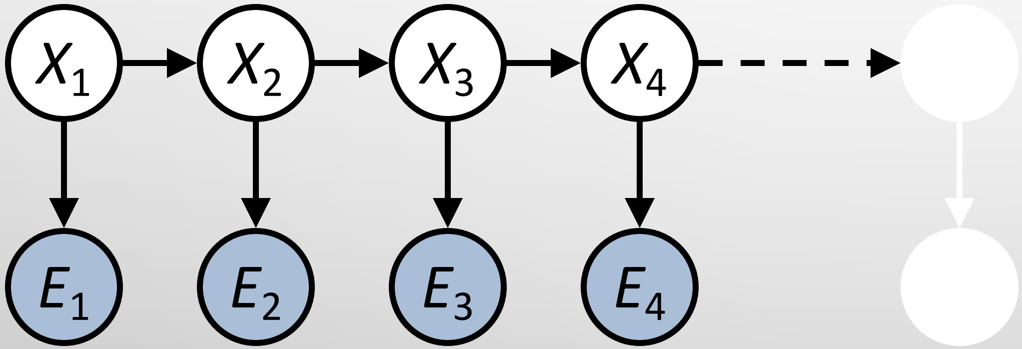


Hidden Markov Models

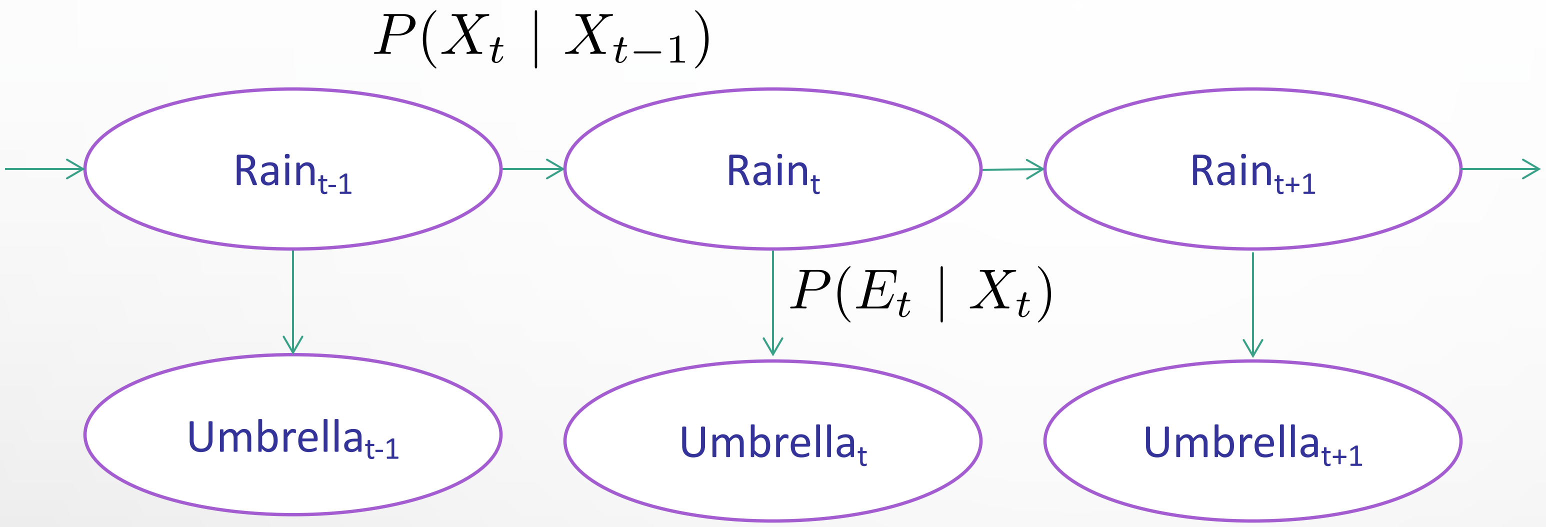
Hidden Markov Models

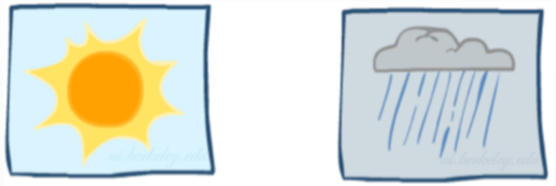
- Markov chains not so useful for most agents
  - Need observations to update your beliefs

- Hidden Markov models (HMMs)
  - Underlying Markov chain over states X
  - You observe outputs (effects) at each time step



Example: Weather HMM



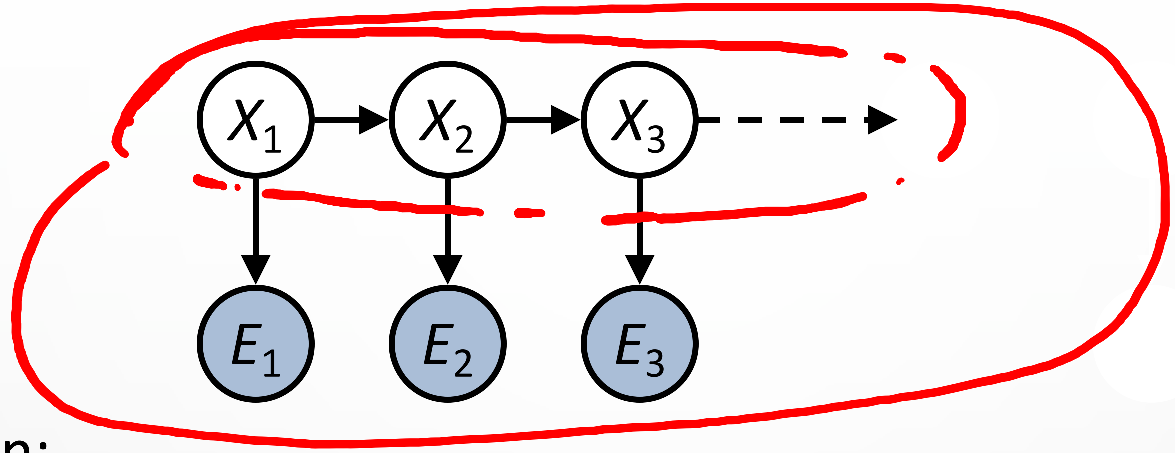


- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t \mid X_{t-1})$
  - Emissions: $P(E_t \mid X_t)$

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:

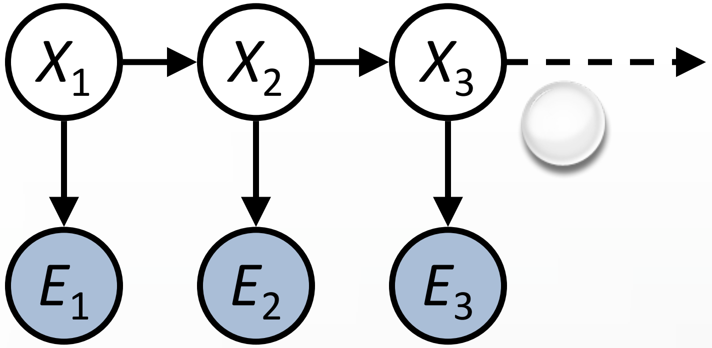
$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3)$$

- More generally:

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

- Questions to be resolved:
  - Does this indeed define a joint distribution?
  - Can every joint distribution be factored this way, or are we making some assumptions about the joint distribution by using this factorization?

Chain Rule and HMMs



- From the chain rule, *every* joint distribution over $X_1, E_1, X_2, E_2, X_3, E_3$ can be written as:

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

- *Assuming* that

$$X_2 \perp\!\!\!\perp E_1 \mid X_1, \quad E_2 \perp\!\!\!\perp X_1, E_1 \mid X_2, \quad X_3 \perp\!\!\!\perp X_1, E_1, E_2 \mid X_2, \quad E_3 \perp\!\!\!\perp X_1, E_1, X_2, E_2 \mid X_3$$

Gives us the expression posited on the previous slide:

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

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)
  - States are positions on a map (continuous)

Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution $B_t(x) = P(X_t \mid E_1, \ldots, E_t)$ (the belief state) over time

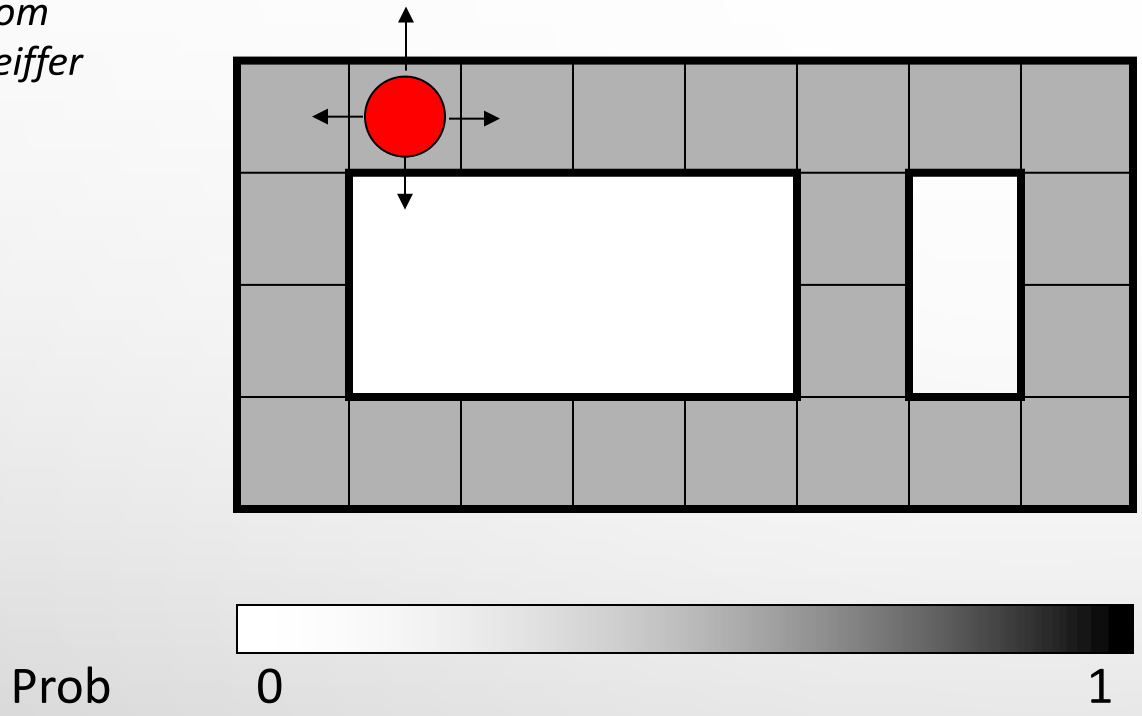
- We start with $B_0(x)$ in an initial setting, usually uniform

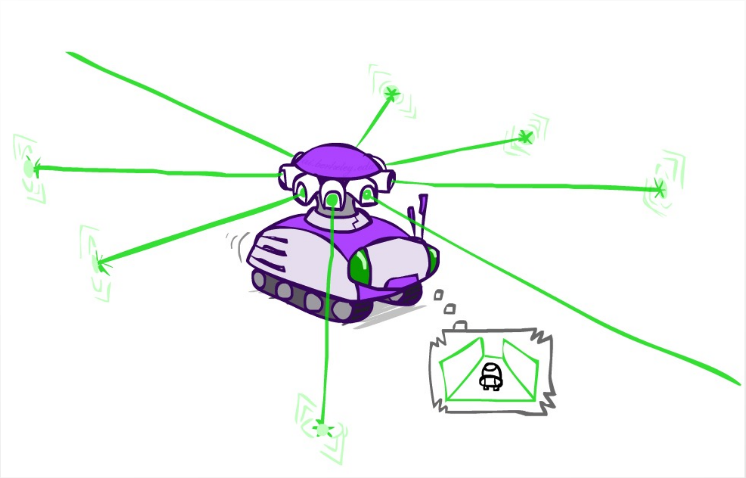
- As time passes, or we get observations, we update $B(x)$

- 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

Example from Michael Pfeiffer



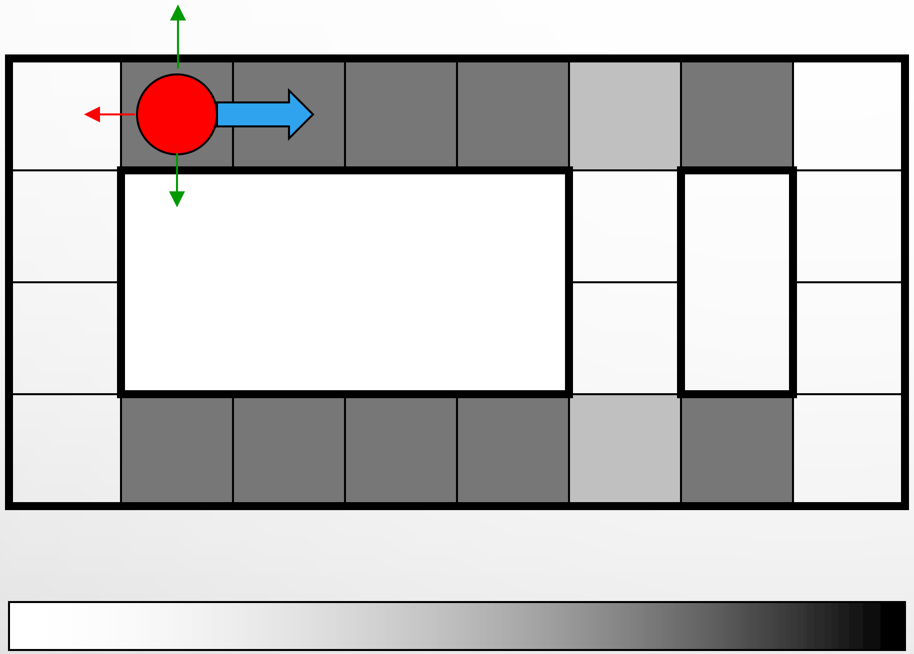


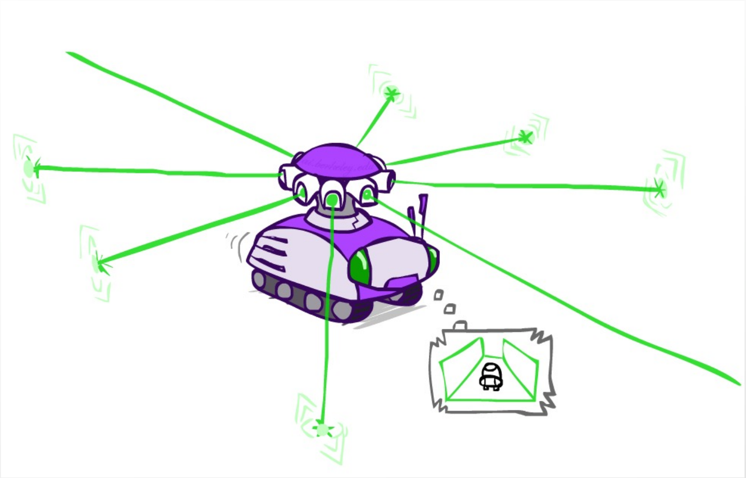
t=0

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



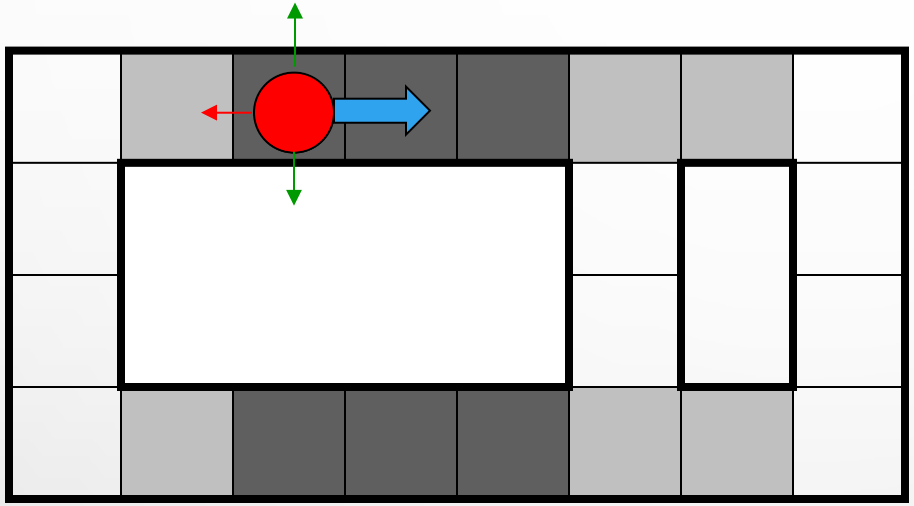


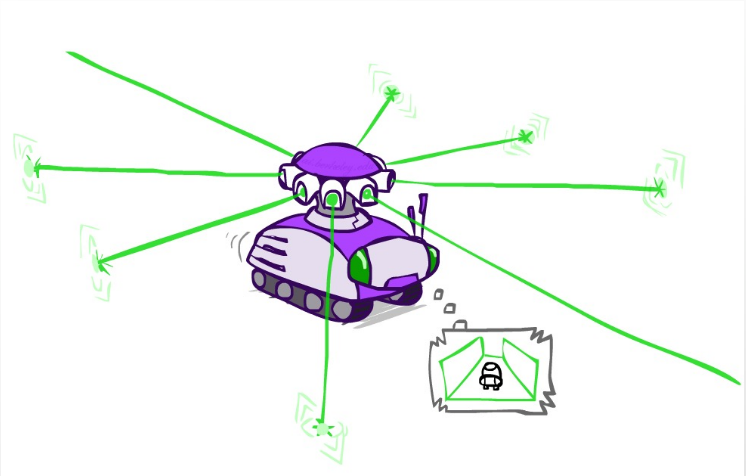
Prob 0 1

t=1

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

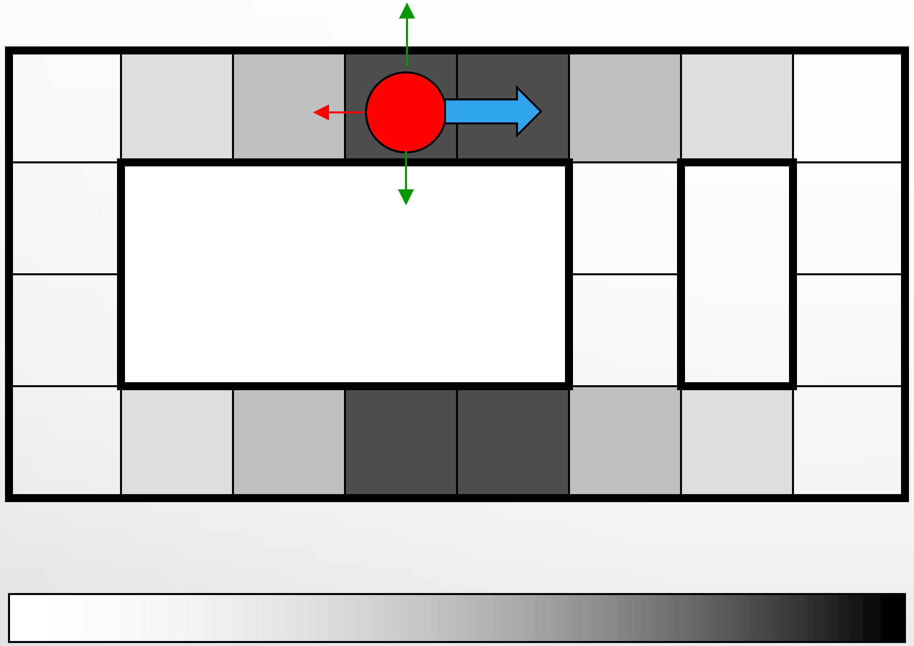
Example: Robot Localization

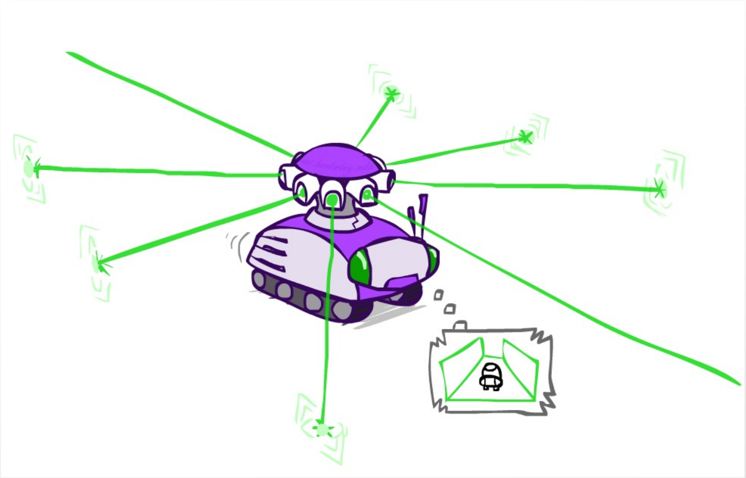




t=2

Example: Robot Localization

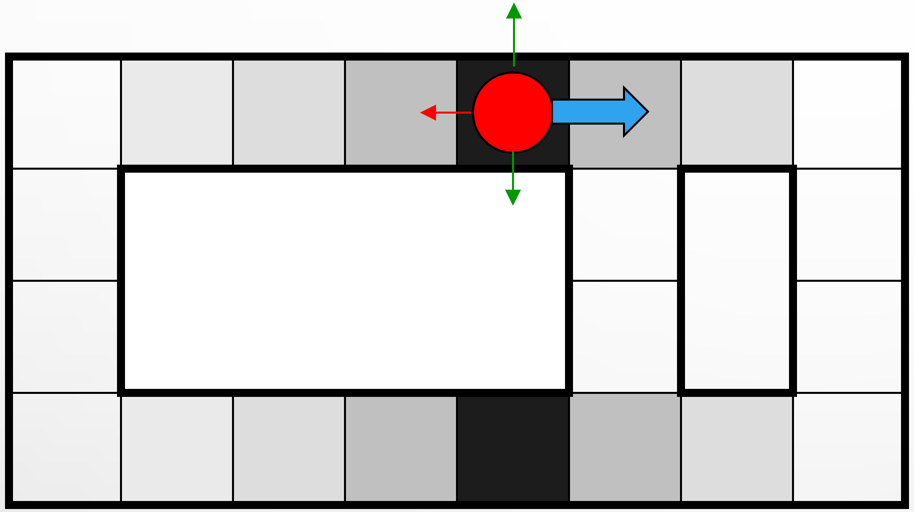


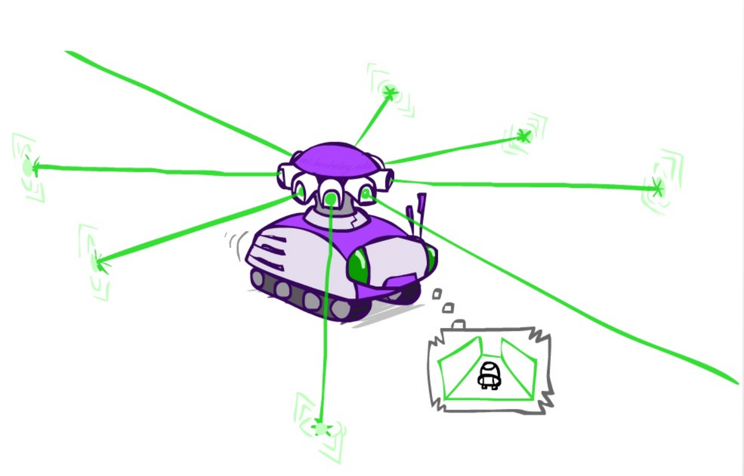


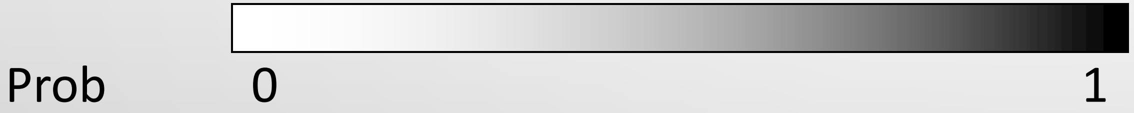
Prob 0 1

t=3

Example: Robot Localization

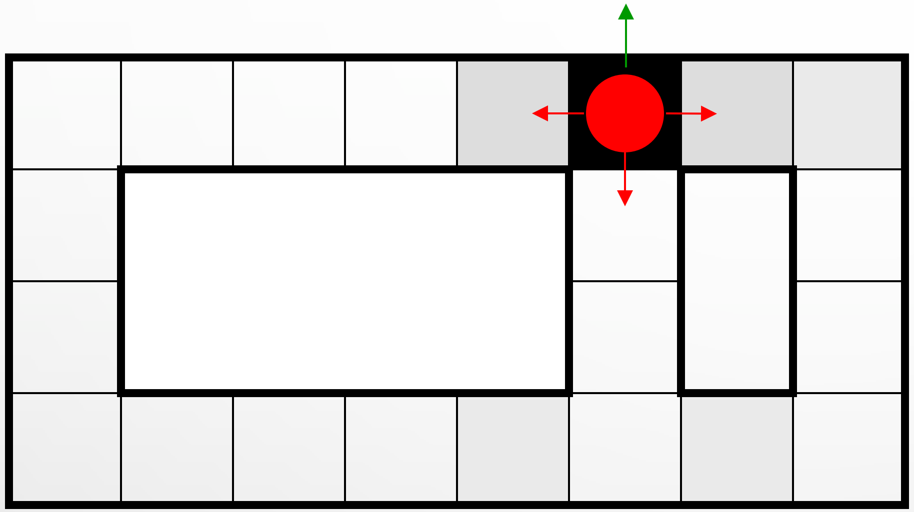


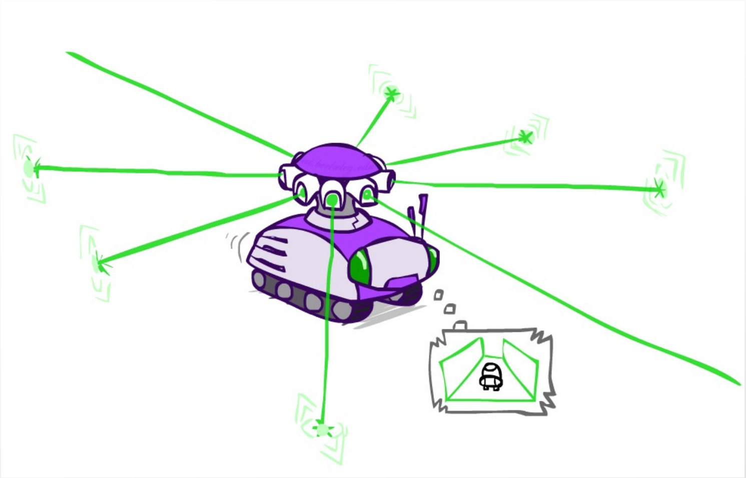




t=4

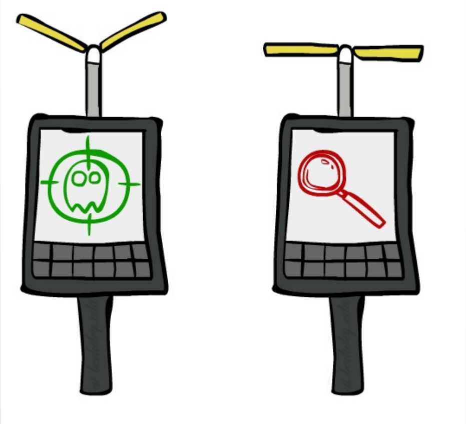
Example: Robot Localization

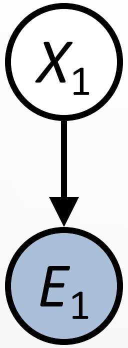


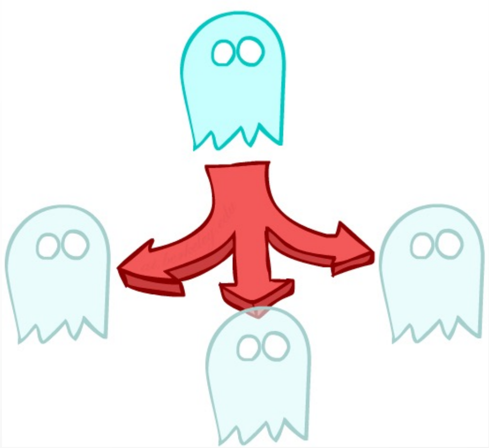


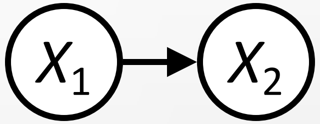
t=5

Inference: Base Cases









$$P(X_1|e_1)$$

$$P(x_1|e_1) = P(x_1, e_1)/P(e_1)$$

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

$$= P(x_1)P(e_1|x_1)$$

$$P(X_2)$$

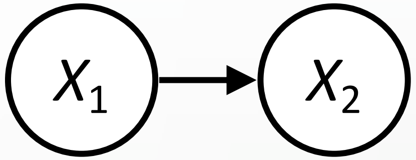
$$P(x_2) = \sum_{x_1} P(x_1, x_2)$$

$$= \sum_{x_1} P(x_1)P(x_2|x_1)$$

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:

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

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

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

- 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

Example: Passage of Time

- As time passes, uncertainty "accumulates"

(Transition model: ghosts usually go clockwise)

<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	1.00	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

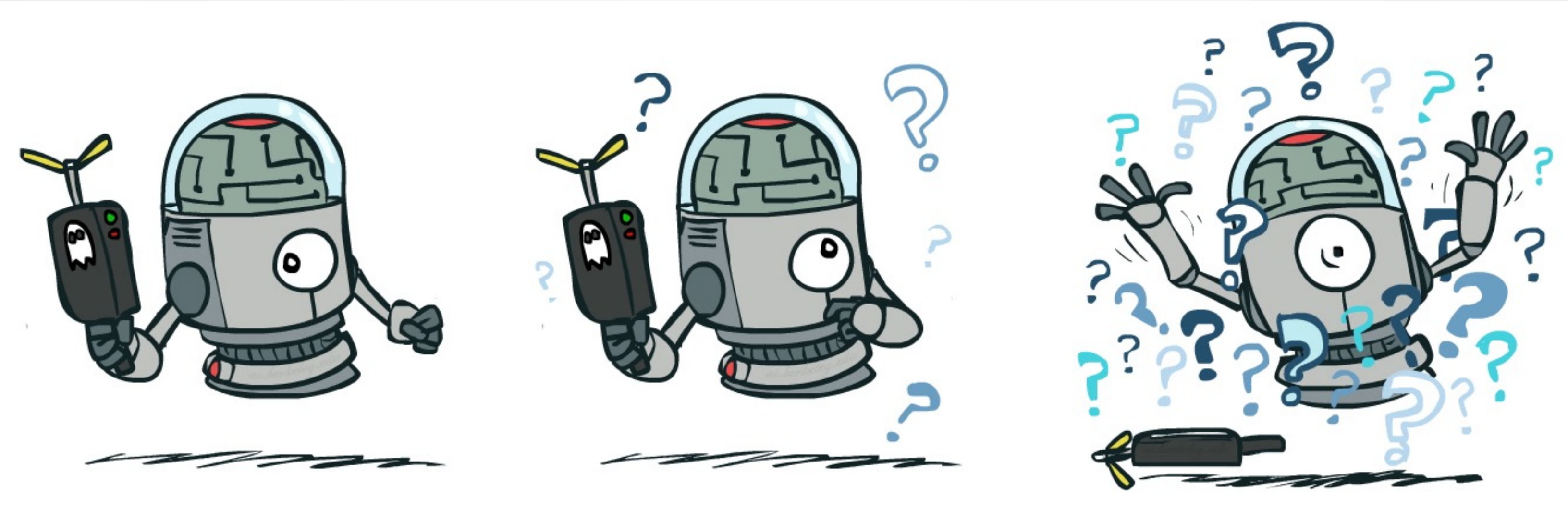
t = 1

<0.01	<0.01	<0.01	<0.01	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01
<0.01	0.76	0.06	0.06	<0.01	<0.01
<0.01	<0.01	0.06	<0.01	<0.01	<0.01

t = 2

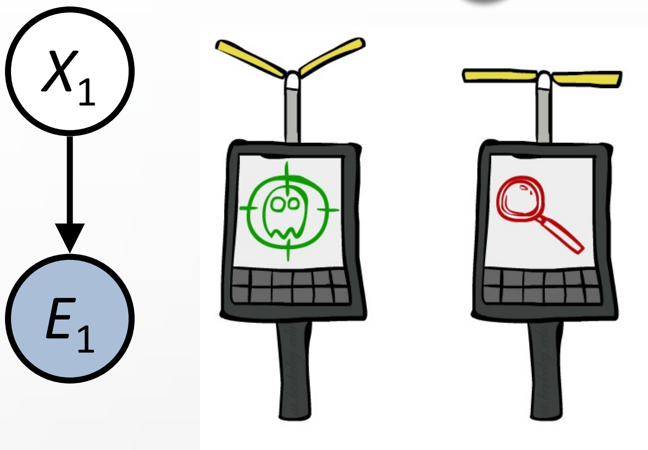
0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

t = 5



Observation

$B_0 \rightarrow B'_1 \rightarrow B_1 \rightarrow B'_2 \rightarrow B_2 \rightarrow \cdots \rightarrow B_t$



- 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:

$$P(X_{t+1}|e_{1:t+1}) = P(X_{t+1}, e_{t+1}|e_{1:t})/P(e_{t+1}|e_{1:t})$$

$$\propto_{X_{t+1}} P(X_{t+1}, e_{t+1}|e_{1:t})$$

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

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

- 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”

0.05	0.01	0.05	<0.01	<0.01	<0.01
0.02	0.14	0.11	0.35	<0.01	<0.01
0.07	0.03	0.05	<0.01	0.03	<0.01
0.03	0.03	<0.01	<0.01	<0.01	<0.01

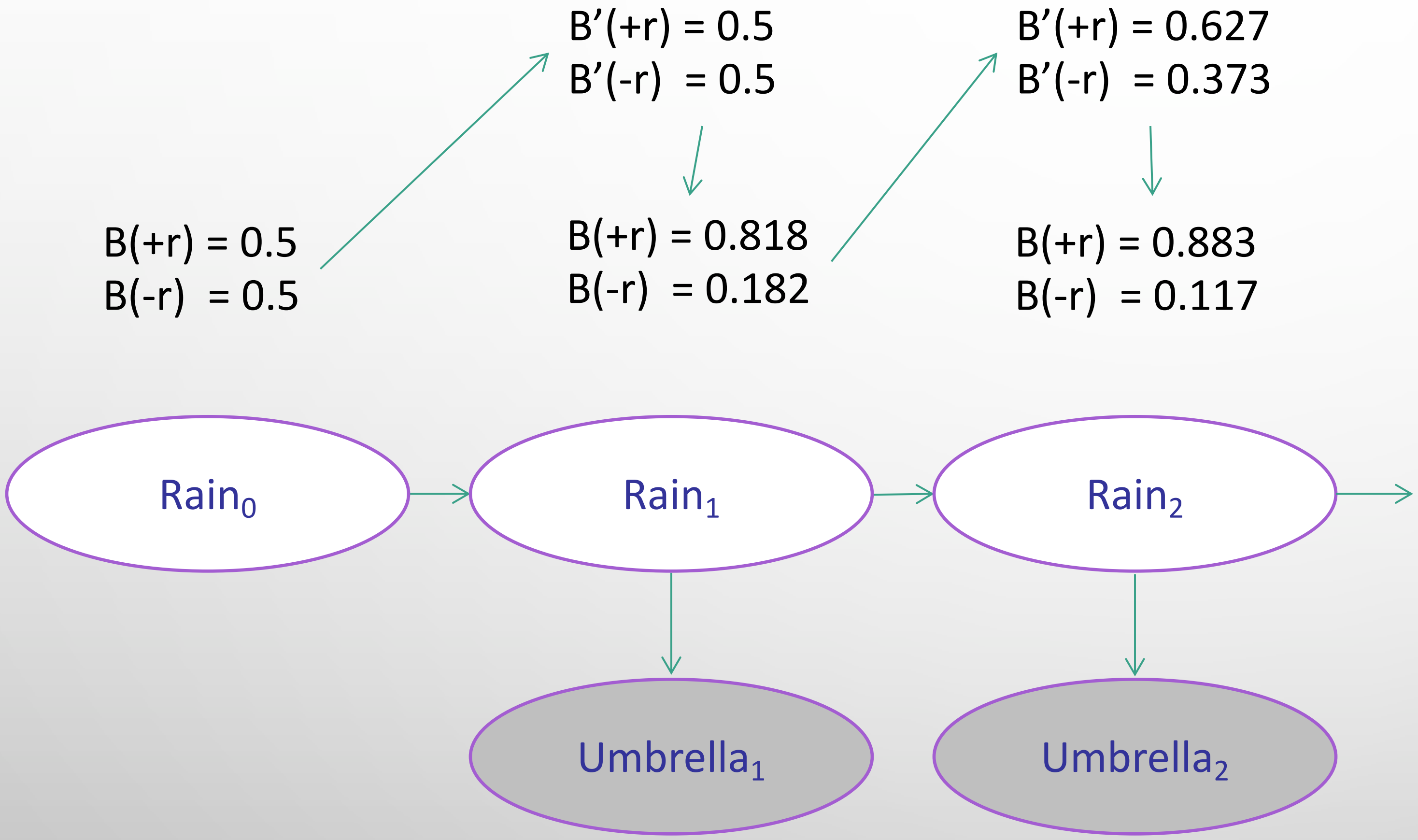
Before observation

<0.01	<0.01	<0.01	<0.01	0.02	<0.01
<0.01	<0.01	<0.01	0.83	0.02	<0.01
<0.01	<0.01	0.11	<0.01	<0.01	<0.01
<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

After observation

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

Example: Weather HMM



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

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

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

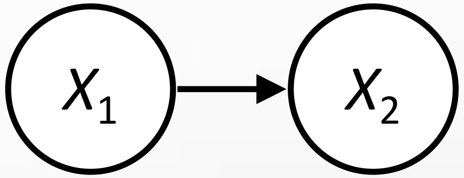
We can normalize as we go if we want to have P(x|e) at each time step, or just once at the end…

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

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

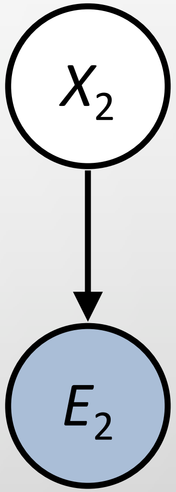
$$= P(e_t | x_t) \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1}, e_{1:t-1})$$

Online Belief Updates



- Every time step, we start with current P(X | evidence)

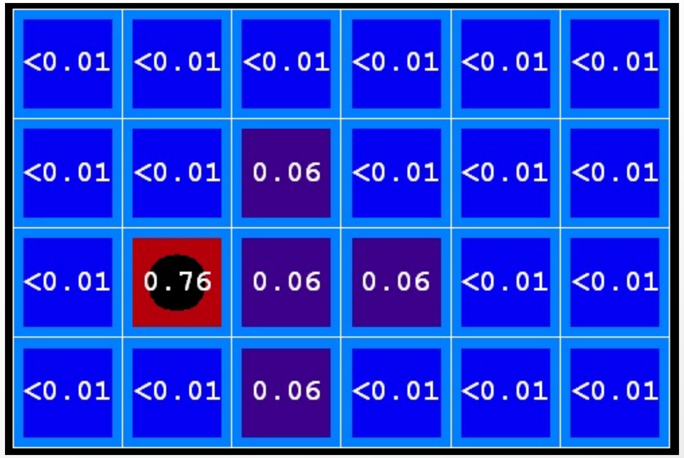
- We update for time: $$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$
- We update for evidence: $$P(x_t|e_{1:t}) \propto_X P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$
- The forward algorithm does both at once (and doesn't normalize)



Recap: Filtering

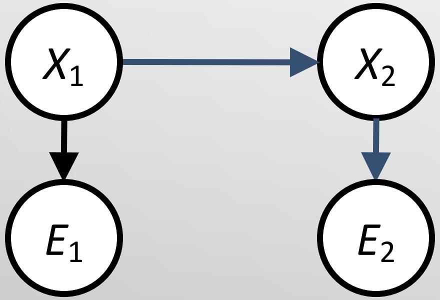
Elapse time: compute P(X_t | $e_{1:t-1}$)

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



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

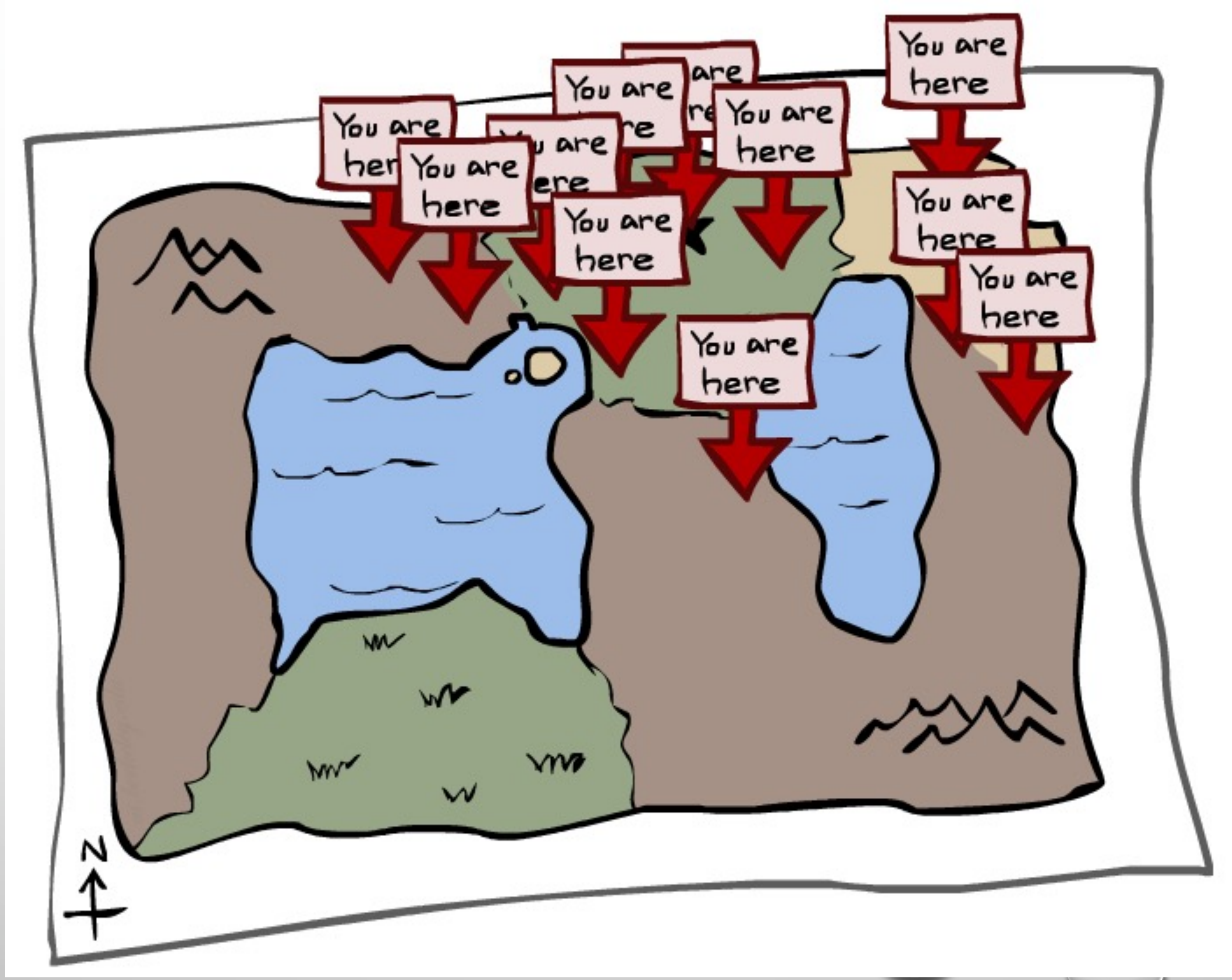
$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$



Belief: <P(rain), P(sun)>

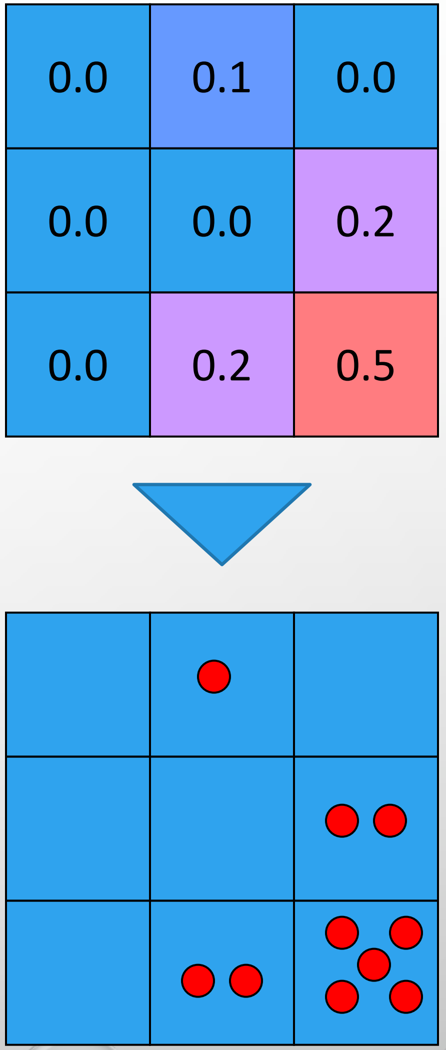
$P(X_1)$	<0.5, 0.5>	*Prior on X_1*
$P(X_1 \mid E_1 = umbrella)$	<0.82, 0.18>	*Observe*
$P(X_2 \mid E_1 = umbrella)$	<0.63, 0.37>	*Elapse time*
$P(X_2 \mid E_1 = umb, E_2 = umb)$	<0.88, 0.12>	*Observe*

Particle Filtering



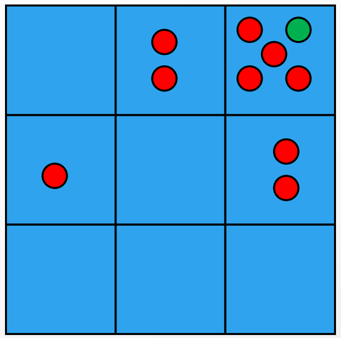
Particle Filtering

- Filtering: approximate solution
- 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



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
  - So, many x may have p(x) = 0!
  - More particles, more accuracy
- For now, all particles have a weight of 1



Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)

Particle Filtering: Elapse Time

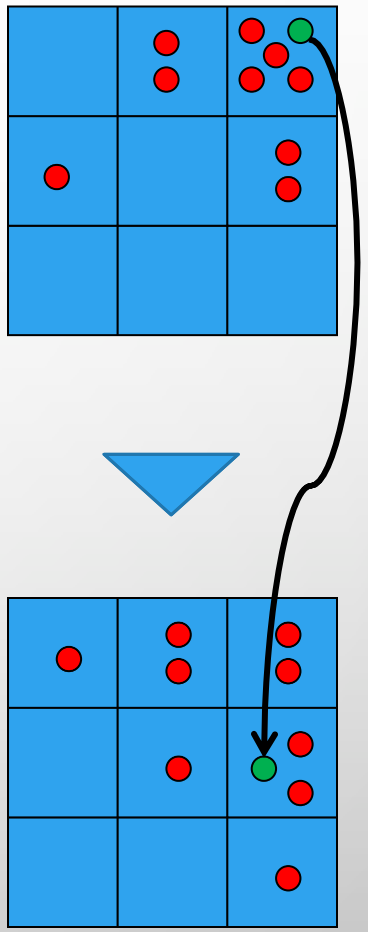
- Each particle is moved by sampling its next position from the transition model

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

  - 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 enough samples, close to exact values before and after (consistent)

Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(1,2)
(3,3)
(3,3)
(2,3)



Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

Particle Filtering: Observe

- Slightly trickier:
  - Don't sample observation, fix it
  - Similar to likelihood weighting, downweight samples based on the evidence

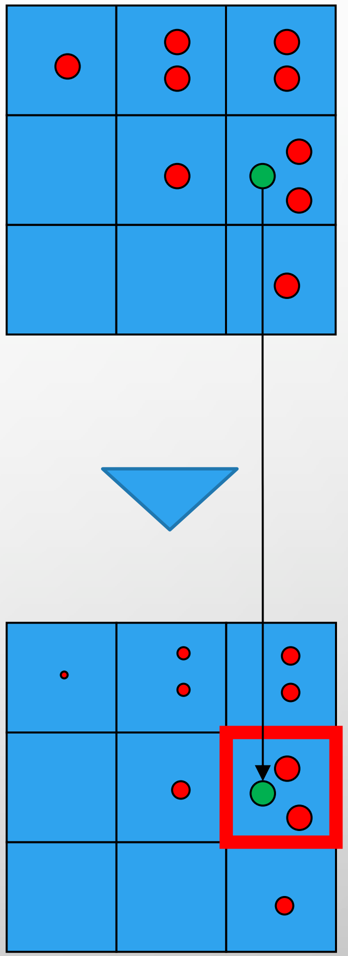
$$w(x) = P(e|x)$$

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

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

Particles:
(3,2)
(2,3)
(3,2)
(3,1)
(3,3)
(3,2)
(1,3)
(2,3)
(3,2)
(2,2)

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

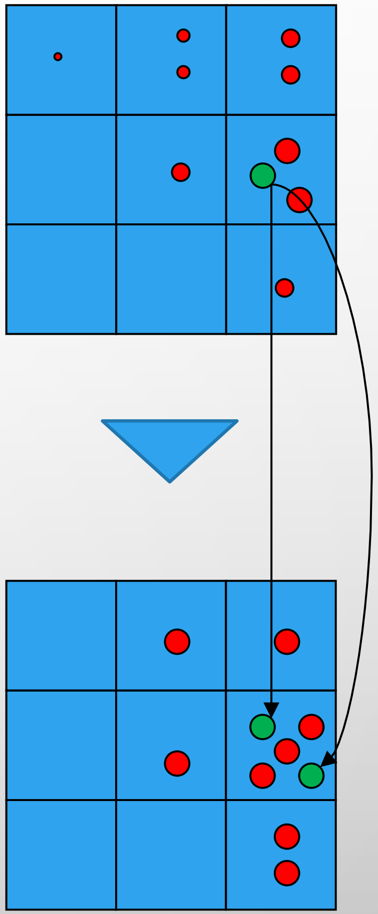


Particle Filtering: Resample

- 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

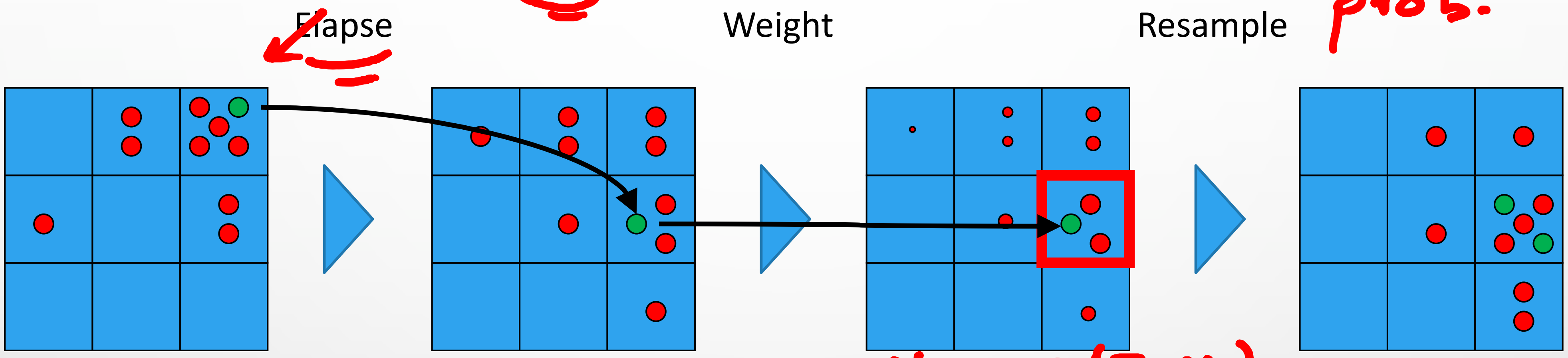
Particles:
(3,2) w=.9
(2,3) w=.2
(3,2) w=.9
(3,1) w=.4
(3,3) w=.4
(3,2) w=.9
(1,3) w=.1
(2,3) w=.2
(3,2) w=.9
(2,2) w=.4

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



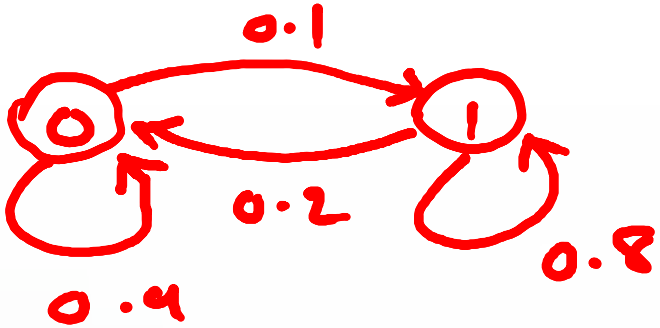
Recap: Particle Filtering

- 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)

Eg. HMM



Trans. prob.

$\mathbb{P}(x_t \mid x_{t-1}=0)$

0	0.9
1	0.1

$\mathbb{P}(x_t \mid x_{t-1}=1)$

0	0.2
1	0.8

Emission prob.

$$\begin{cases} \mathbb{P}(E=2 \mid X=0) = 0.9 \\ \mathbb{P}(E=3 \mid X=0) = 0.1 \end{cases}$$

$$\begin{cases} \mathbb{P}(E=2 \mid X=1) = 0.3 \\ \mathbb{P}(E=3 \mid X=1) = 0.7 \end{cases}$$

$\mathbb{P}(X_5 \mid 2, 2, 3, 2, 2)$

$E_1 \quad E_2 \quad \cdots \quad E_5$ 2   3

Sol. $t=0$ 0, 0, 1

elapse time

$t=1$ 0   1   1

weighting

0.9   0.3   0.3 $\rightarrow \mathbb{P}(X_1 \mid E_1)$

$\left(\frac{0.9}{1.5}, \frac{0.3}{1.5}, \frac{0.3}{1.5}\right)$

3 Sample from this distribution

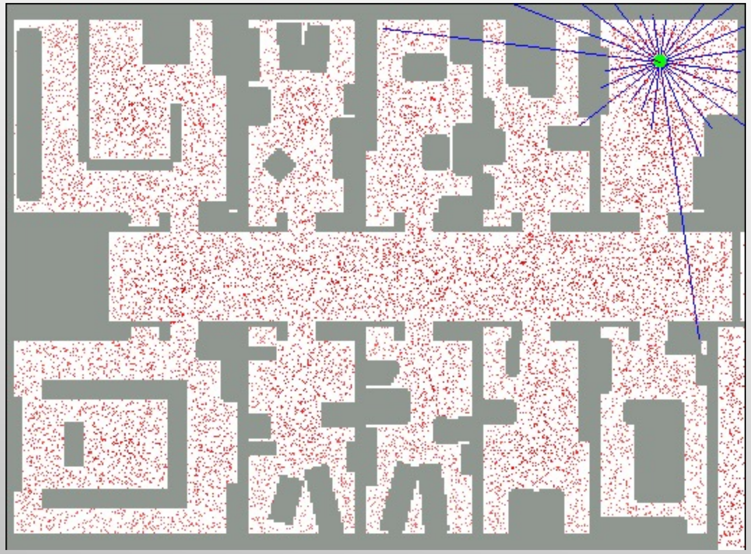
0   0   1

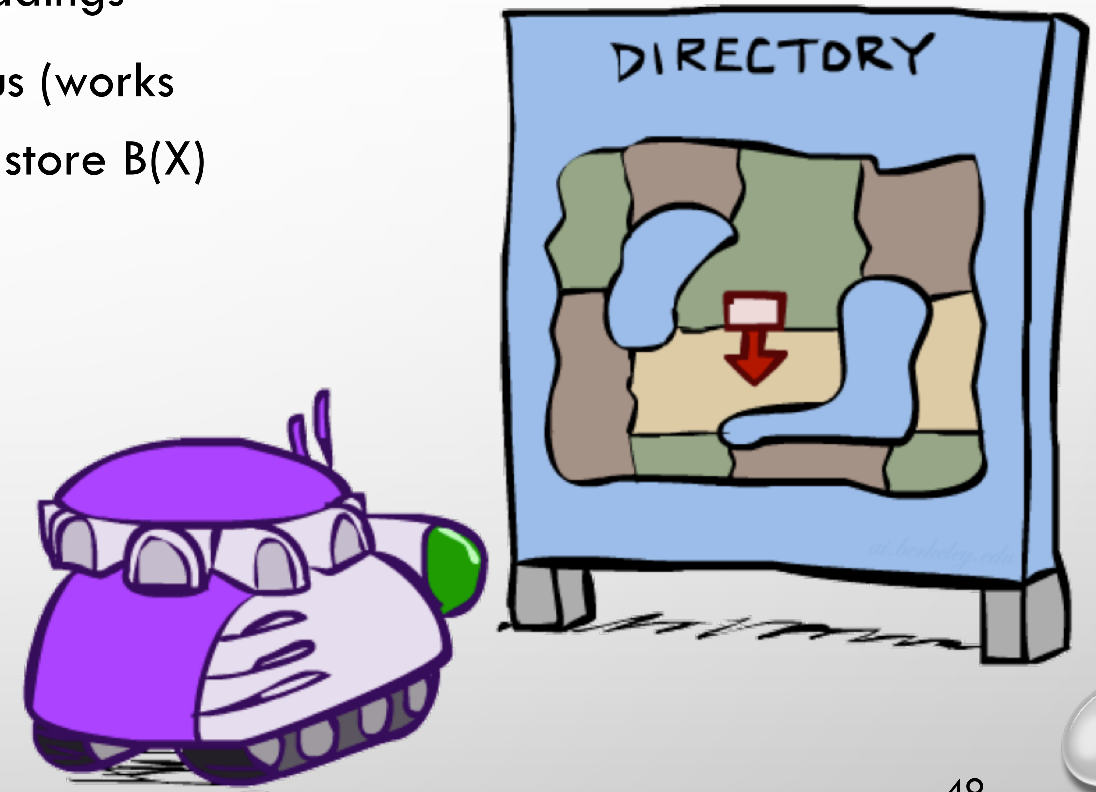
$$(0, 1, 0)$$

$$\begin{cases} \mathbb{P}(X_s = 0 \mid \cdots) \approx 2/3 \\ \mathbb{P}(X_s = 1 \mid \cdots) \approx 1/3 \end{cases}$$

Robot Localization

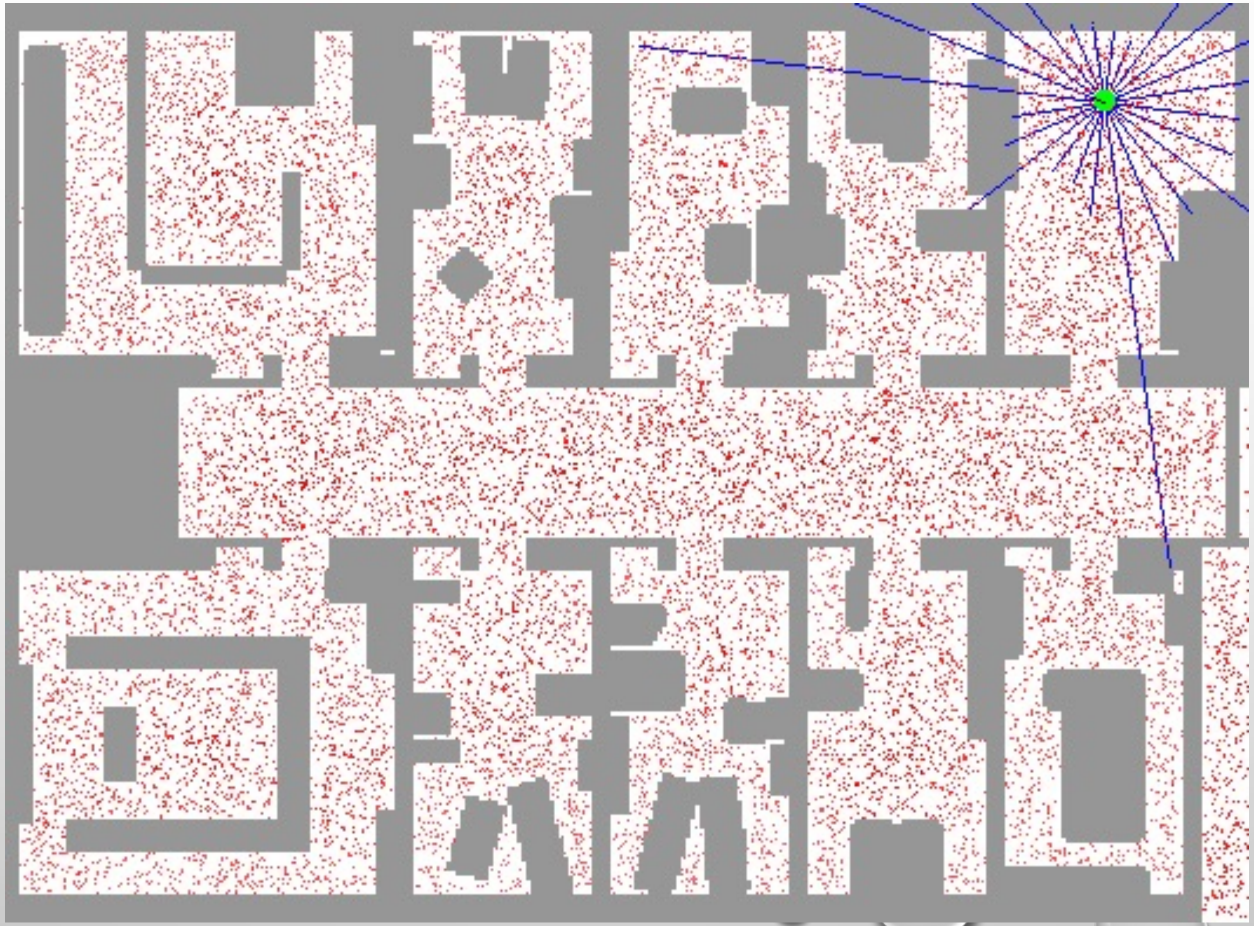
- In robot localization:
  - We know the map, but not the robot's position
  - Observations may be vectors of range finder readings
  - State space and readings are typically continuous (works basically like a very fine grid) and so we cannot store B(X)
  - Particle filtering is a main technique





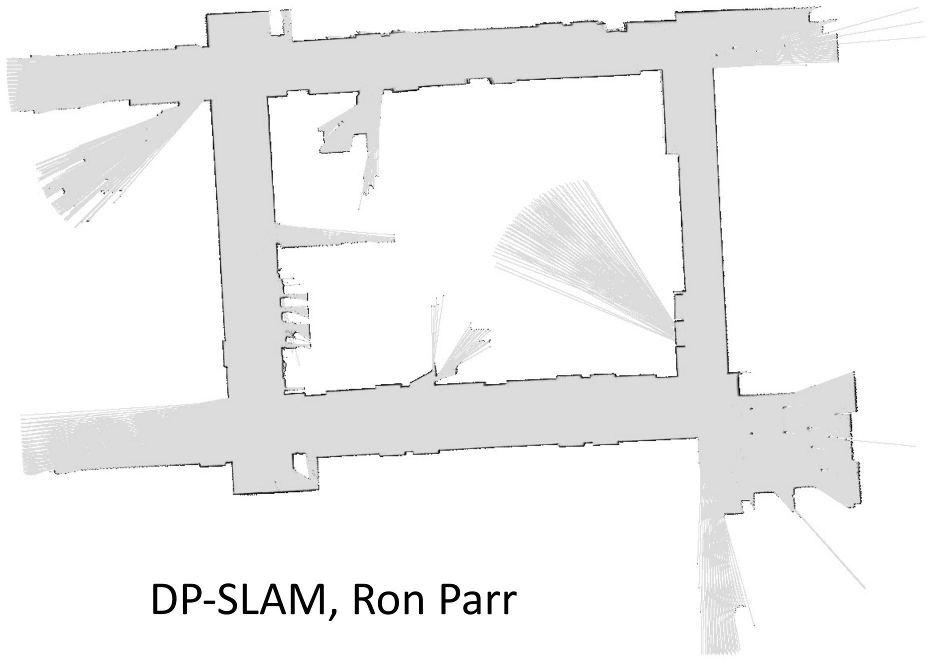
Particle Filter Localization (Sonar)

Particle Filter Localization (Laser)

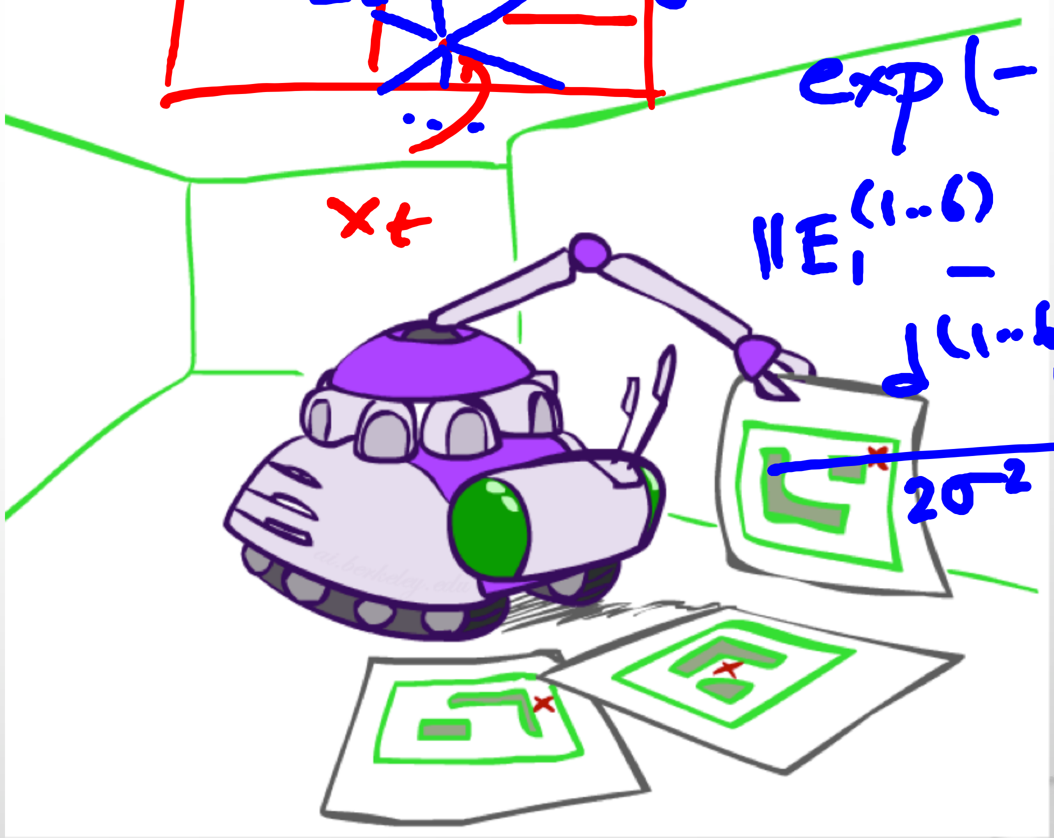


Robot Mapping

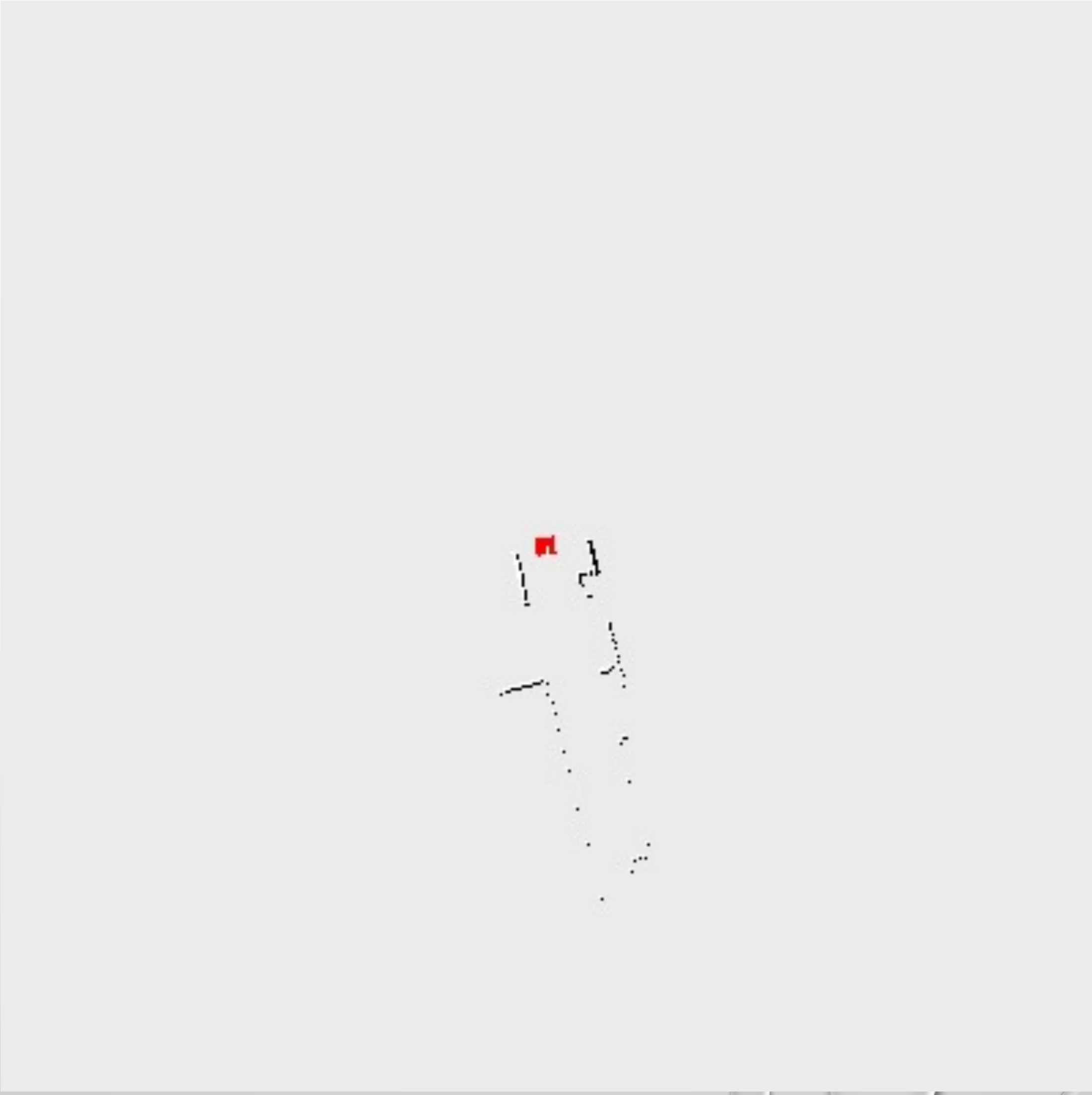
- SLAM: simultaneous localization and mapping
  - We do not know the map or our location
  - State consists of position AND map!
  - Main techniques: Kalman filtering (Gaussian HMMs) and particle methods



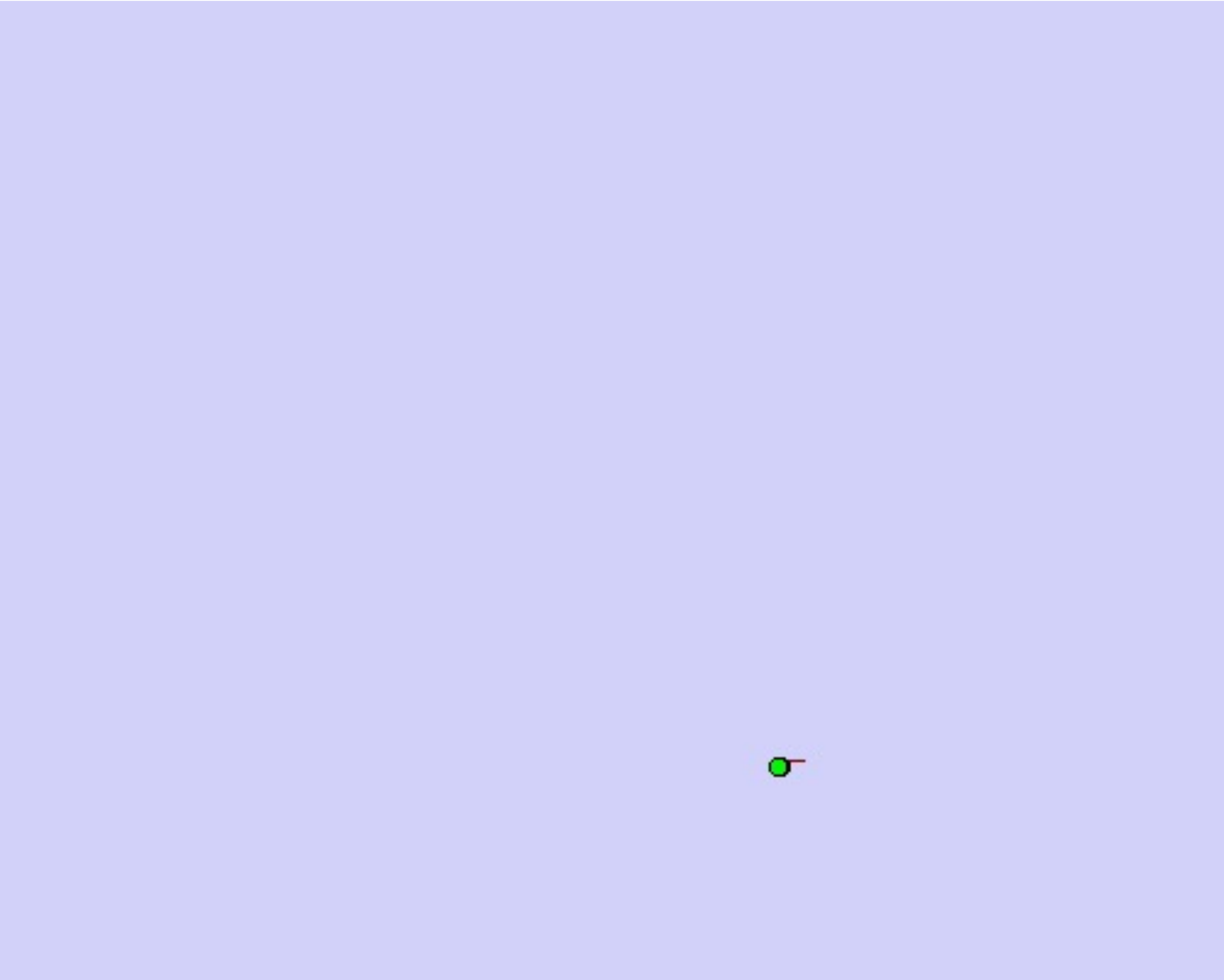
DP-SLAM, Ron Parr



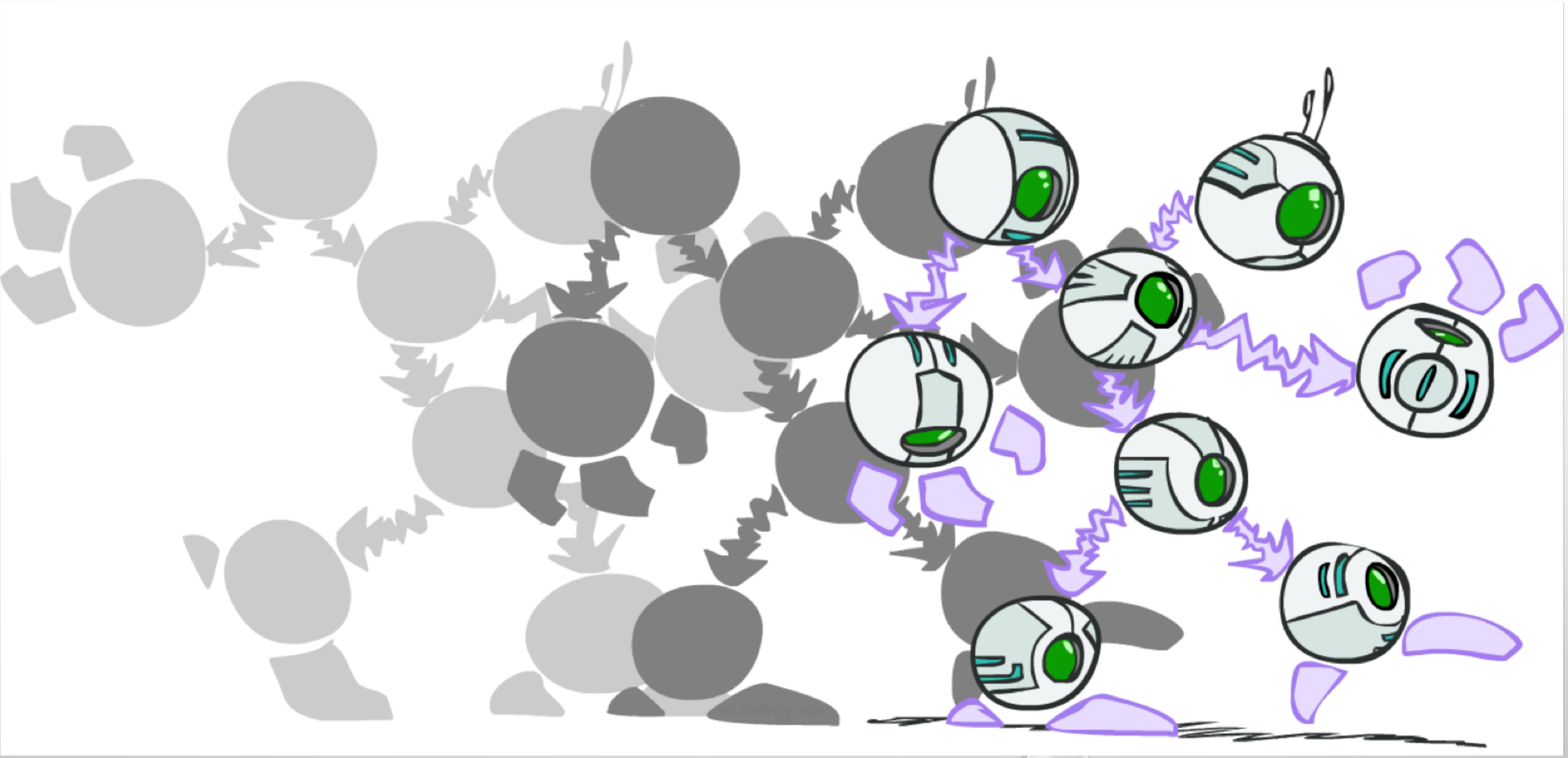
Particle Filter SLAM – Video 1



Particle Filter SLAM – Video 2

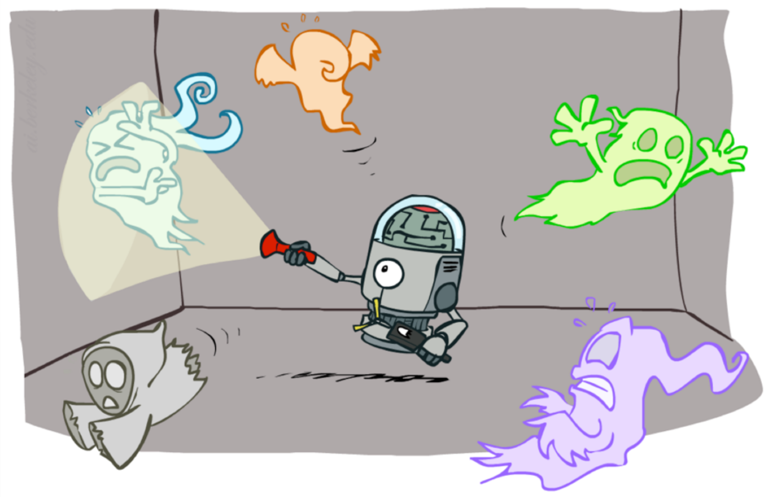


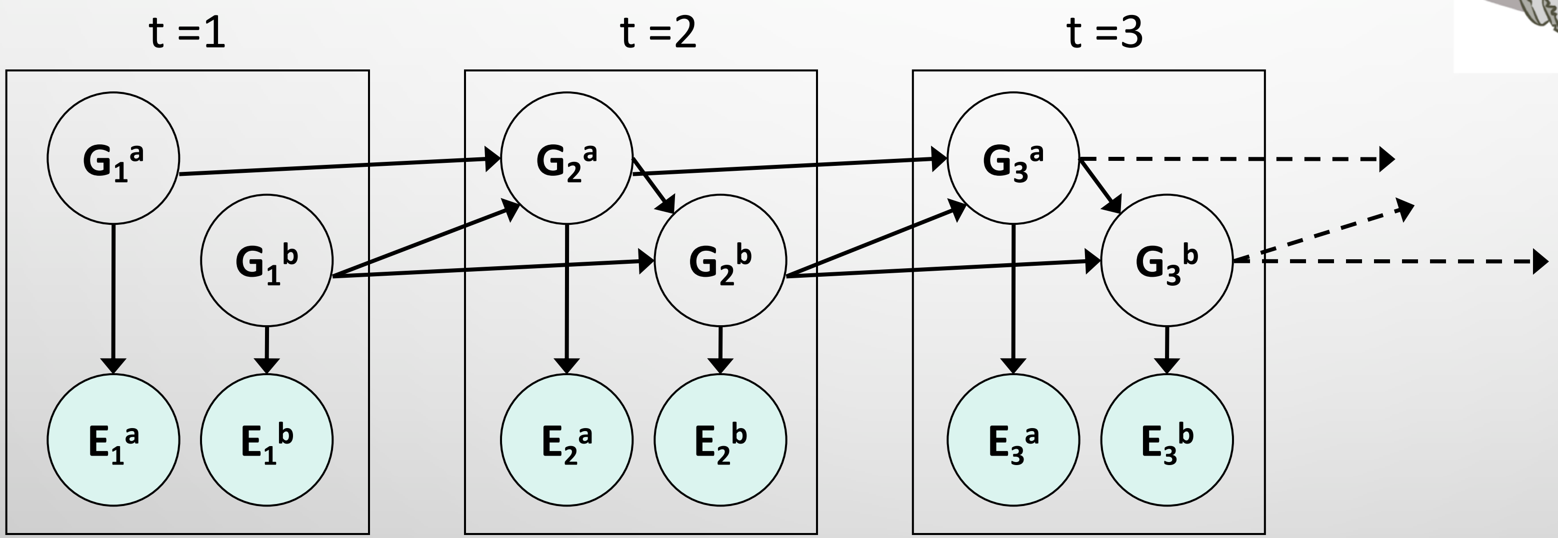
Dynamic Bayes Nets



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

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 *entire* 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

Most Likely Explanation

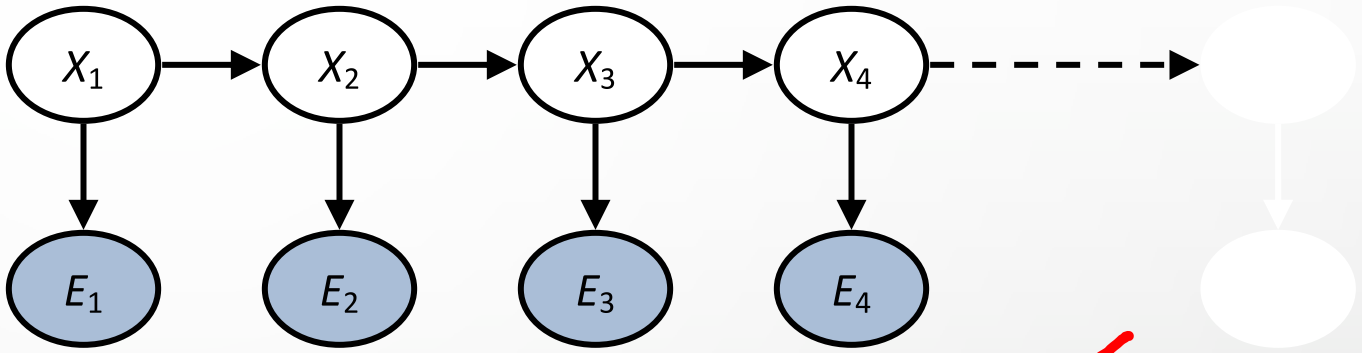
$\mathbb{P}(X_t \mid e_{1..t})$

filtering prob.

$\max_{x_1 \dots x_t} \mathbb{P}(x_1 \dots x_t \mid e_{1..t})$

HMMs: MLE 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

$\propto \mathbb{P}(x_{1..t}, e_{1..t})$

$\rightarrow \mathbb{P}(x_1)\, \mathbb{P}(e_1|x_1)\,\cdot$

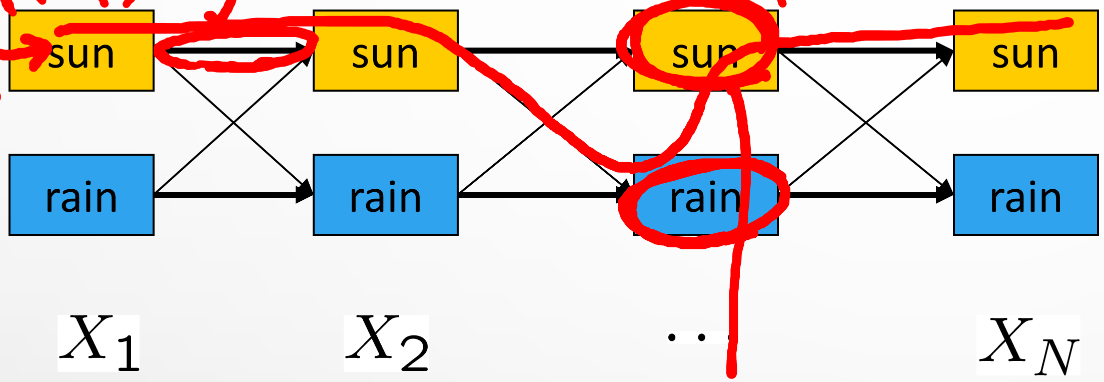
$P(x_{1..t-1}|e_{1..t-1})\, P(x_2|x_1)\, \mathbb{P}(e_2|x_2)$

$\vdots$

$\mathbb{P}(x_t|x_{t-1})\, \mathbb{P}(e_t|x_t)$

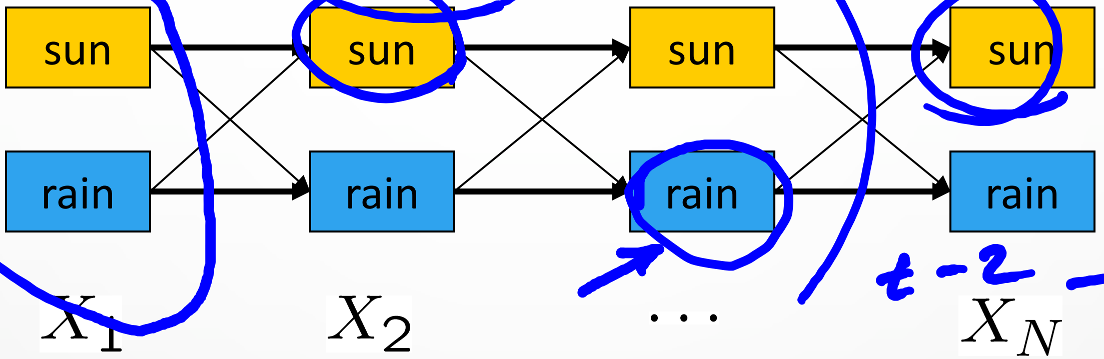
State Trellis

- State trellis: graph of states and transitions over time



- Each arc represents some transition $x_{t-1} \rightarrow x_t$
- 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 / Viterbi Algorithms



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)

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

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

Challenge

$$\max_{X_1 \dots X_t} P(X_1 \dots X_t \mid E_1 \dots E_t)$$

- Setting
  - User we want to spy on use HTTPS to browse the internet
- Measurements
  - IP address
  - Sizes of packets coming in
- Goal
  - Infer browsing sequence of that user
- e.g.: Medical, financial, legal, …

X_t = # web site @ time t

↓

E_t = Size of the packet @ time t

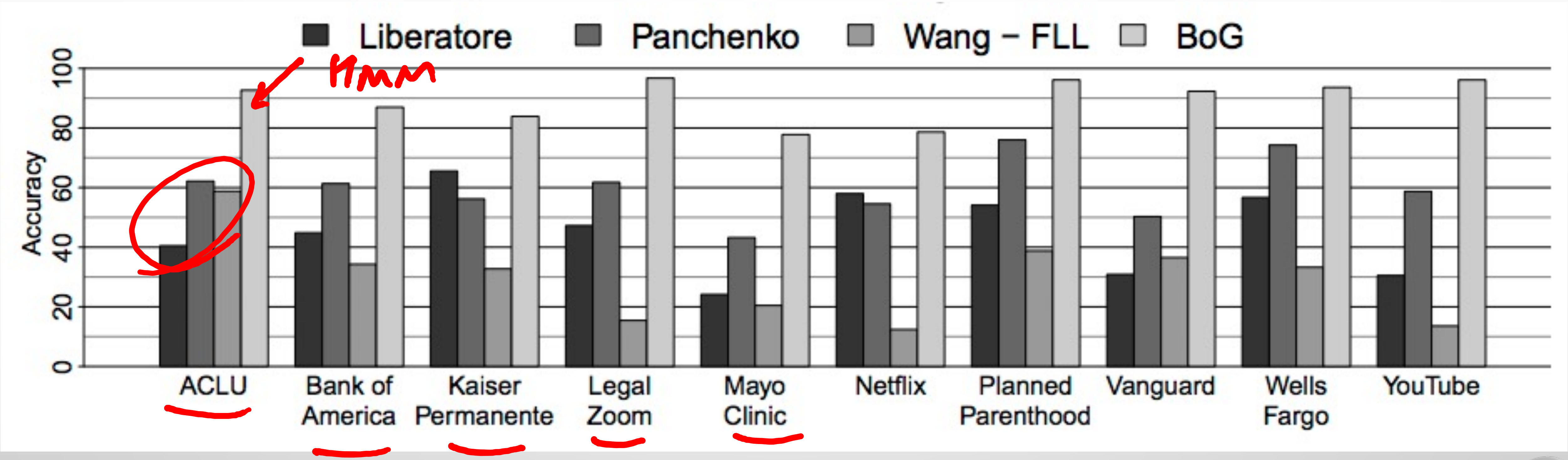
HMM

- Transition model
  - Probability distribution over links on the current page + some probability to navigate to any other page on the site

- Noisy observation model due to traffic variations
  - Caching
  - Dynamically generated content
  - User-specific content, including cookies

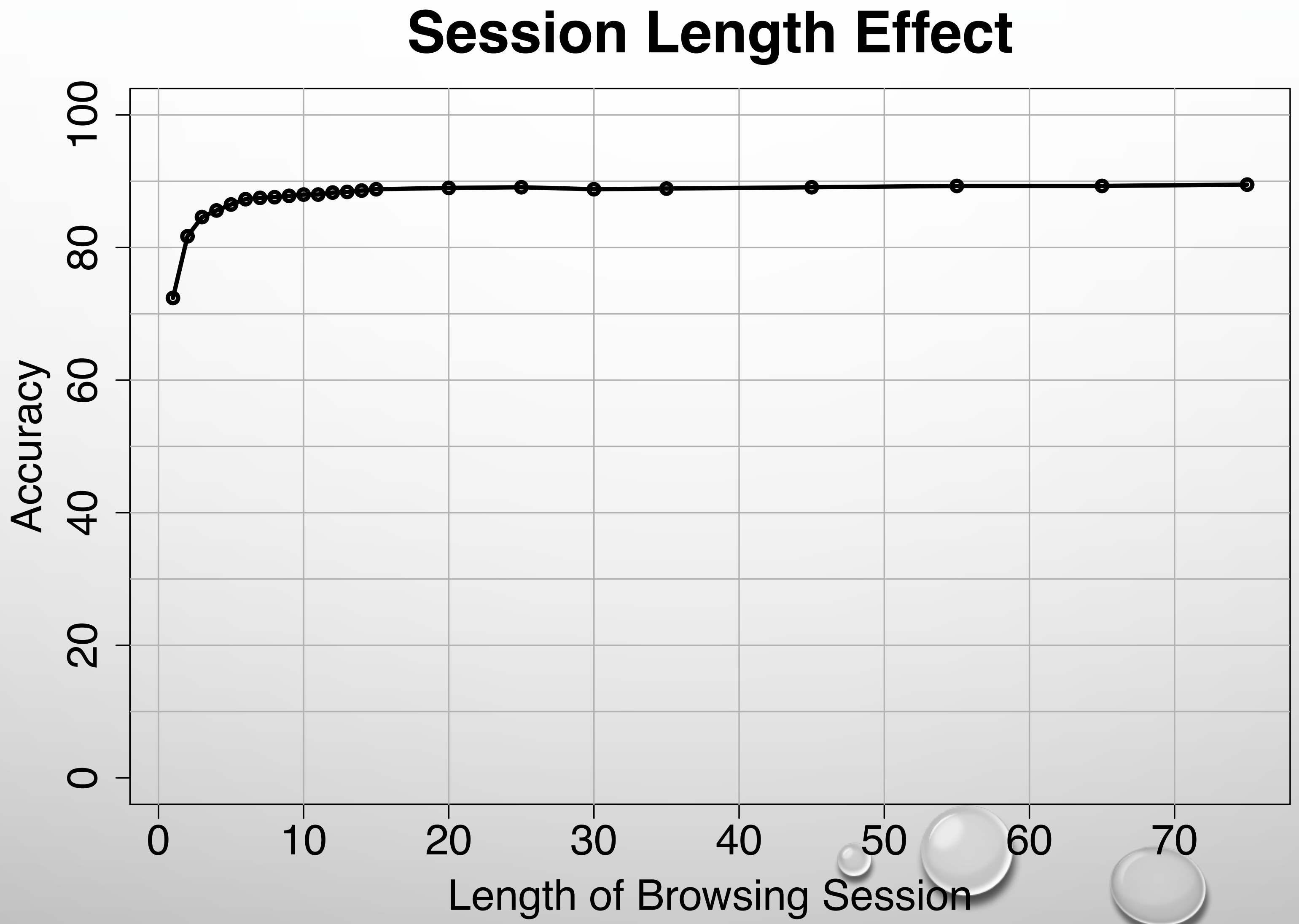
→ Probability distribution P(packet size | page)

Results



BoG = described approach, others are prior work

Results



Session Length Effect
Accuracy
0
20
40
60
80
100
0
10
20
30
40
50
60
70
Length of Browsing Session