343H: Honors AI

Lecture 19:

Probabilistic reasoning over time

4/1/2014

Kristen Grauman

UT Austin

Slides courtesy of Dan Klein, UC Berkeley

Unless otherwise noted

Last time

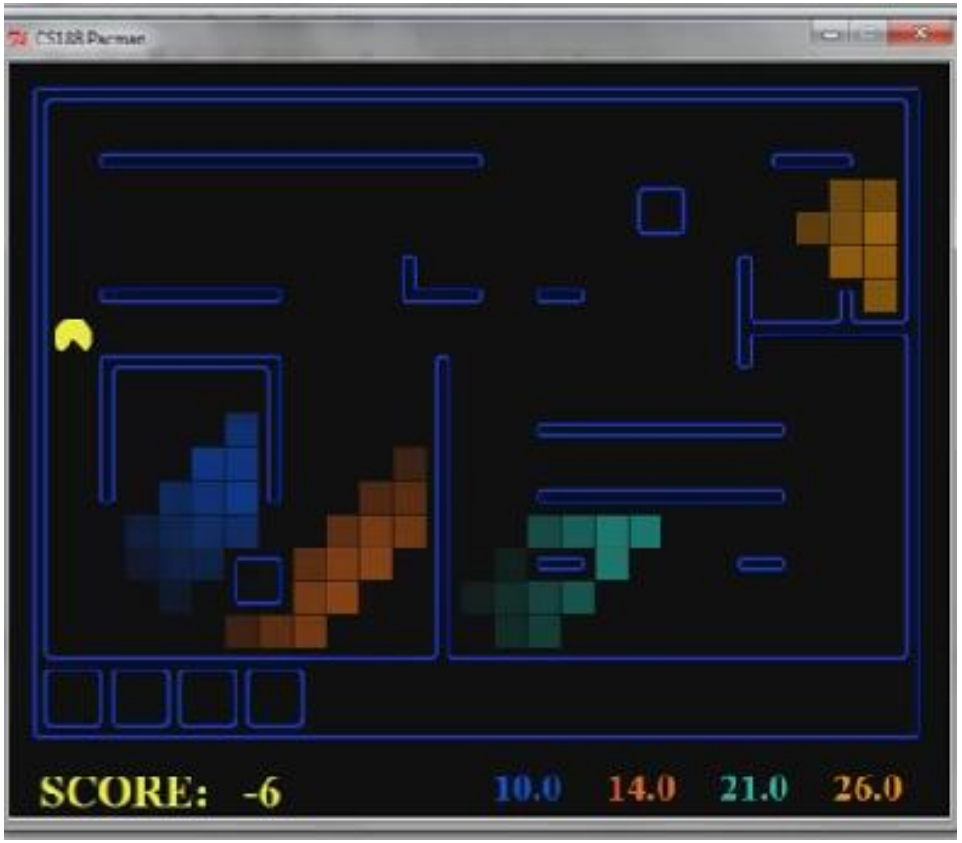
- Decision networks:
  - What action will maximize expected utility?
  - Connection to expectimax
- Value of information:
  - How much are we willing to pay for a sensing action to gather information?

Another VPI example

Today

- HMMs and Particle Filtering

Motivation: invisible ghosts!



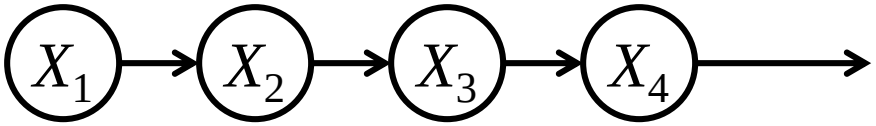
Reasoning over Time or Space

- 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

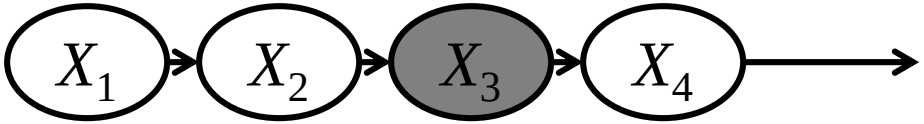
- A Markov model is a chain-structured BN
  - Each node is identically distributed (stationarity)
  - Value of X at a given time is called the state
  - As a BN:



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

  - Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial probs)

Conditional Independence



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

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

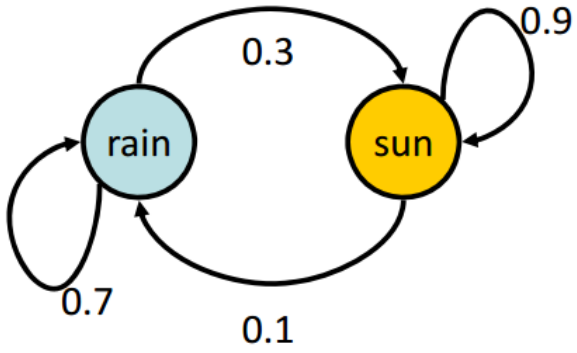
Example Markov Chain: Weather

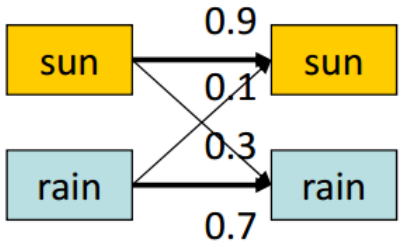
- Weather:
  - States: X = {rain, sun}
  - Initial distribution: 1.0 sun

- 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

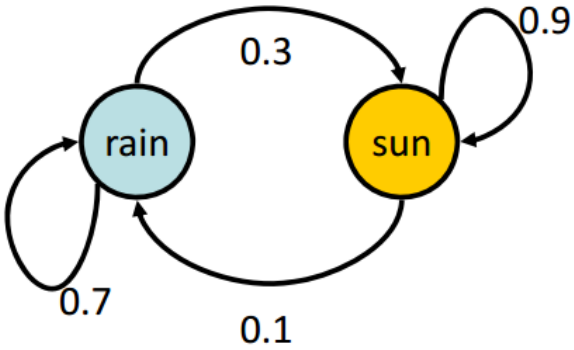
Two new ways of representing the same CPT





Example Markov Chain: Weather

- Initial distribution: 1.0 sun

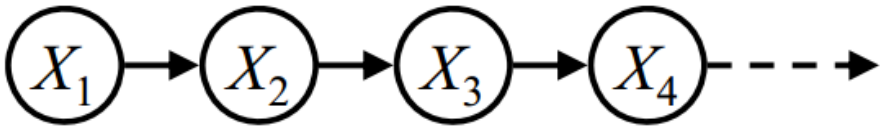


- What is the probability distribution after one step?

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

Mini-Forward Algorithm

- Question: What's P(X) on some day t?
  - An instance of variable elimination! (In order X_1, X_2, …)



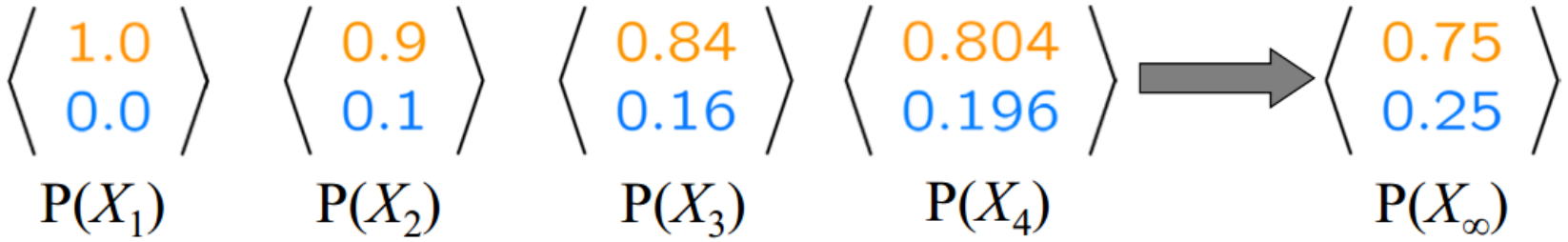
$$P(x_1) = \text{known}$$

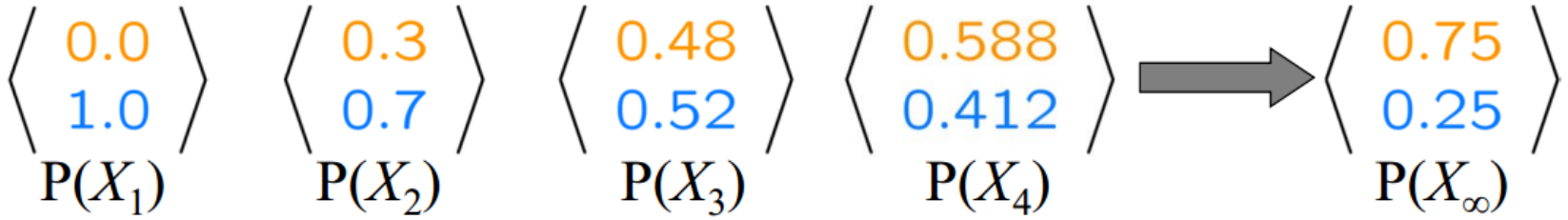
$$P(x_t) = \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1})$$

Forward simulation

Example run of mini-forward algorithm

- From initial observation of sun





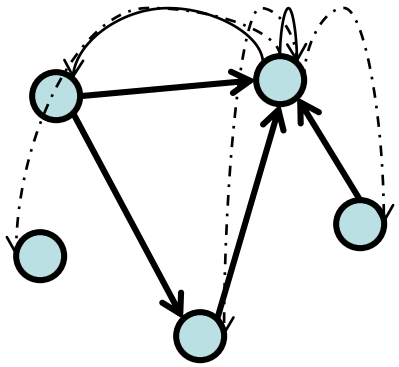
- From yet another initial distribution $P(X_1)$:

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 of the chain
  - It satisfies $P_\infty(X) = P_{\infty+1}(X) = \sum_x P_{t+1|t}(X|x)P_\infty(x)$

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:
  - With probability 1/n resample variable X_j according to

    $$P(X_j \mid x_1, x_2, \ldots, x_{j-1}, x_{j+1}, \ldots, x_n, e_1, \ldots, e_m)$$

- Stationary distribution:
  - Conditional distribution $P(X_1, X_2, \ldots, X_n \mid e_1, \ldots, e_m)$
  - Means that when running Gibbs sampling long enough we get a sample from the desired distribution
  - Requires some proof to show this is true!

Markov Chain Example: Text

"A dog is a man's best friend. It's a dog eat dog world out there."

$p(\mathbf{x}_t|\mathbf{x}_{t-1})$

$\mathbf{x}_{t-1}$ \ $\mathbf{x}_t$	a	dog	is	man's	best	friend	it's	eat	world	out	there	.
a		**2/3**		**1/3**								
dog			**1/3**					**1/3**	**1/3**			
is	**1**											
man's					**1**							
best						**1**						
friend												**1**
it's	**1**											
eat		**1**										
world										**1**		
out											**1**	
there												**1**
.							**1**					

Source: S. Seitz

Text synthesis

Create plausible looking poetry, love letters, term papers, etc.

Most basic algorithm

1. Build probability histogram
   - find all blocks of N consecutive words/letters in training documents
   - compute probability of occurrence $p(\mathbf{x}_t|\mathbf{x}_{t-1},\ldots,\mathbf{x}_{t-(n-1)})$
2. Given words $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_{k-1}$
   - compute $\mathbf{x}_k$ by sampling from $p(\mathbf{x}_t|\mathbf{x}_{t-1},\ldots,\mathbf{x}_{t-(n-1)})$

Source: S. Seitz

Markov Random Field

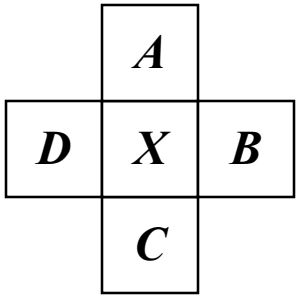
A Markov random field (MRF)

- generalization of Markov chains to two or more dimensions.

First-order MRF:

- probability that pixel *X* takes a certain value given the values of neighbors *A*, *B*, *C*, and *D*:

$$P(\mathrm{X}|\mathrm{A},\mathrm{B},\mathrm{C},\mathrm{D})$$



Source: S. Seitz

Texture Synthesis [Efros & Leung, ICCV 99]

Can apply 2D version of text synthesis

Texture corpus (sample)

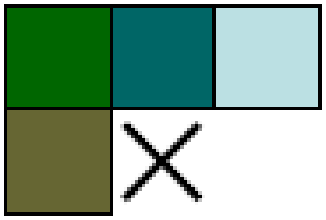
Output

Texture synthesis: intuition

Before, we inserted the next word based on existing nearby words…

Now we want to insert **pixel intensities** based on existing nearby pixel values.

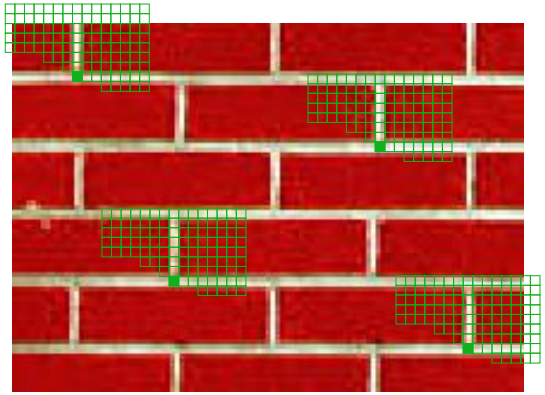
Sample of the texture ("corpus")



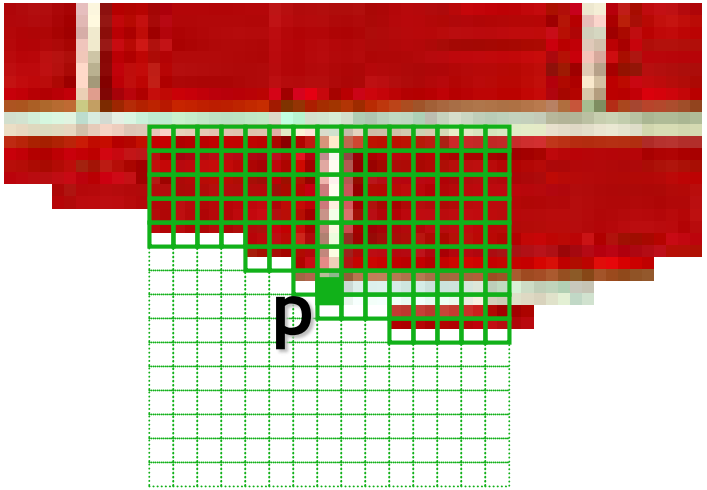
Place we want to insert next

Distribution of a value of a pixel is conditioned on its neighbors alone.

Synthesizing One Pixel



input image

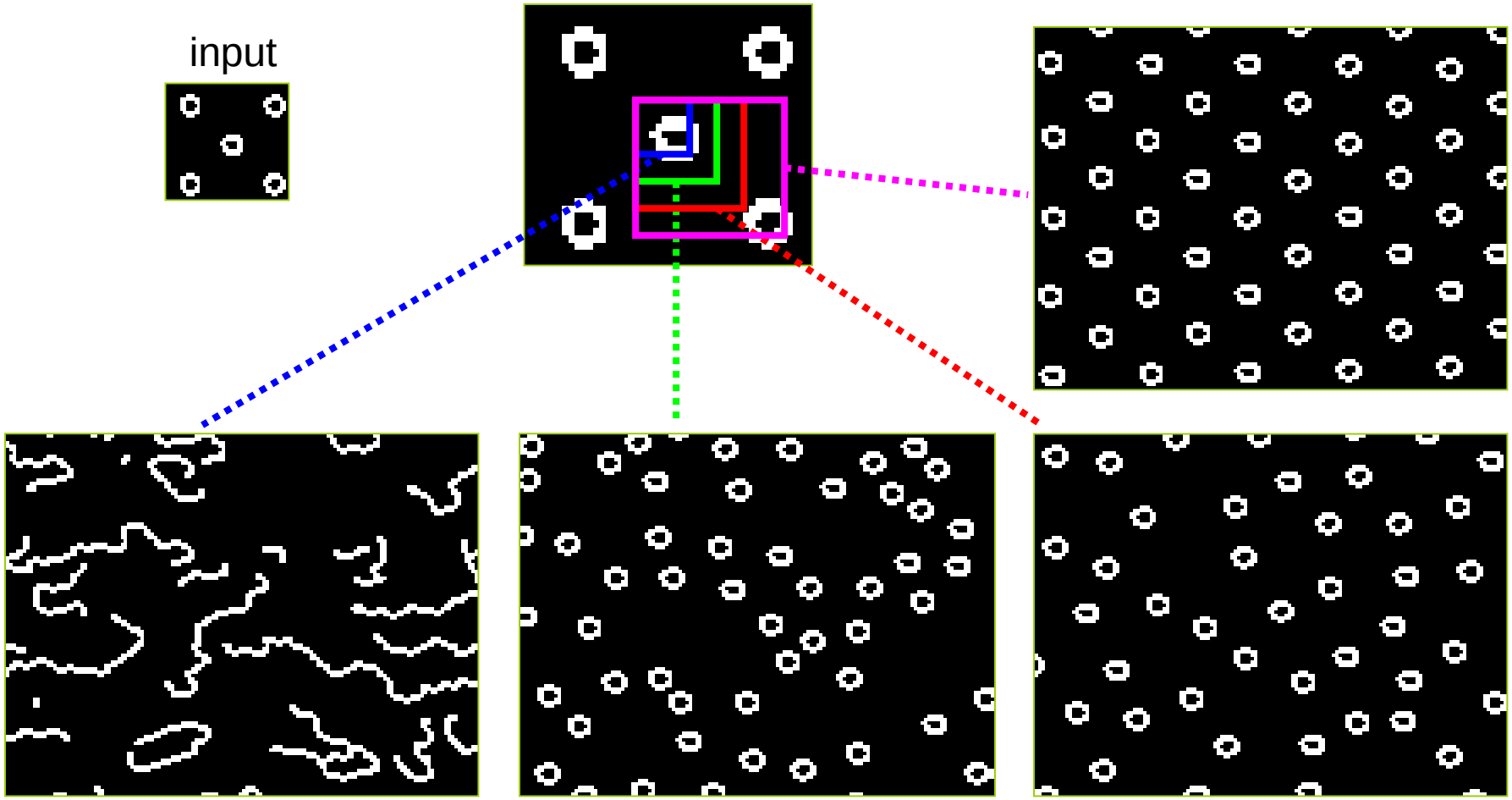


synthesized image

- What is $P(\mathbf{x}|\text{neighborhood of pixels around x})$?
- Find all the windows in the image that match the neighborhood
- To synthesize **x**
  - pick one matching window at random
  - assign **x** to be the center pixel of that window

Slide from Alyosha Efros, ICCV 1999

Neighborhood Window

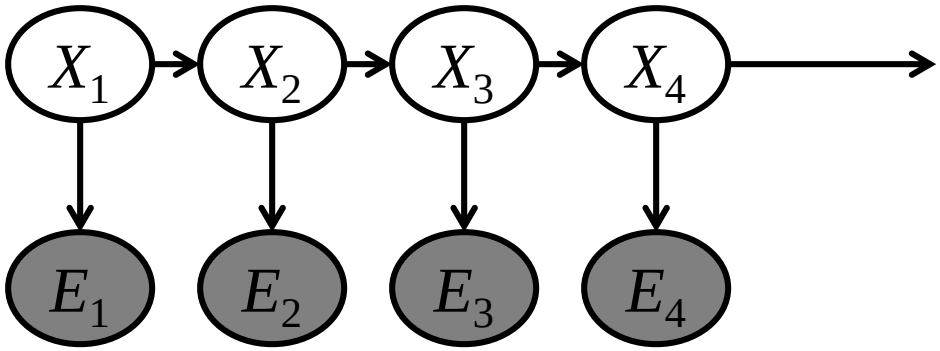


Slide from Alyosha Efros, ICCV 1999

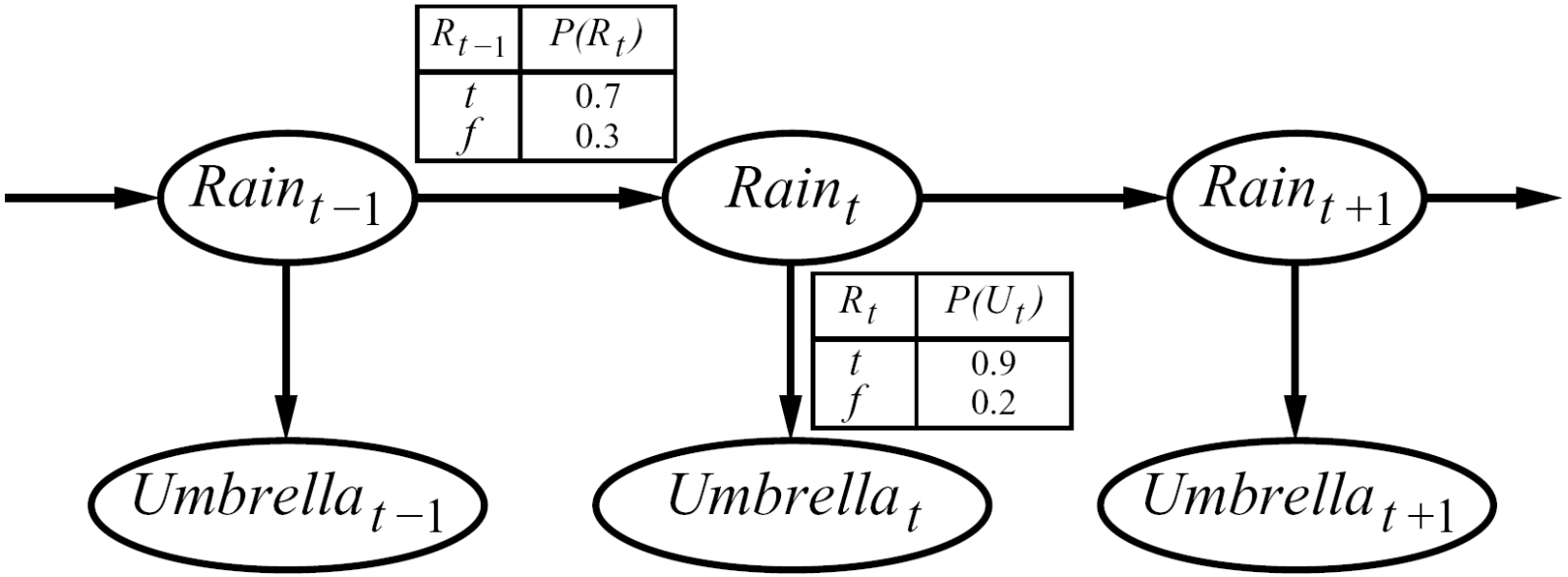
Hidden Markov Models

- Markov chains not so useful for most agents
  - Eventually you don't know anything anymore
  - Need observations to update your beliefs

- Hidden Markov models (HMMs)
  - Underlying Markov chain over states S
  - You observe outputs (effects) at each time step
  - As a Bayes' net:



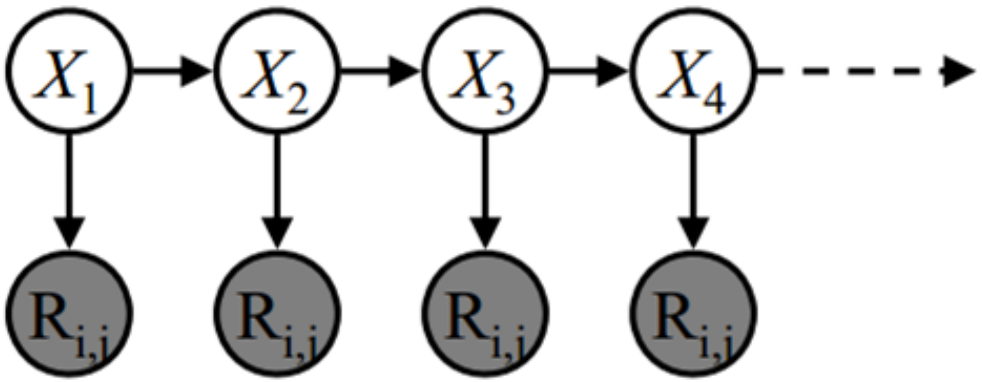
Example: Weather HMM



- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X|X_{-1})$
  - Emissions: $P(E|X)$

Example: Ghostbusters HMM

- $P(X_1)$ = uniform
- $P(X|X')$ = usually move clockwise, but sometimes move in a random direction or stay in place
- $P(R_{ij}|X)$ = same sensor model as before: red means close, green means far away.



1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

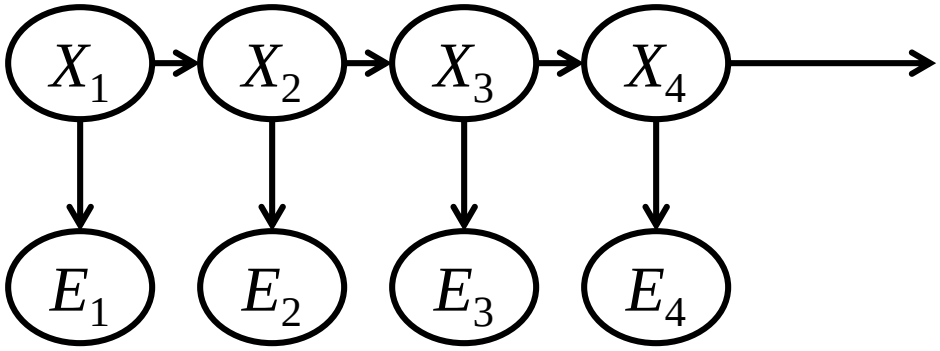
$P(X_1)$

1/6	1/6	1/2
0	1/6	0
0	0	0

$P(X|X' =<1,2>)$

Conditional Independence

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



- Quiz: does this mean that evidence variables guaranteed to be independent?
  - [No, correlated by the hidden state]

Real HMM Examples

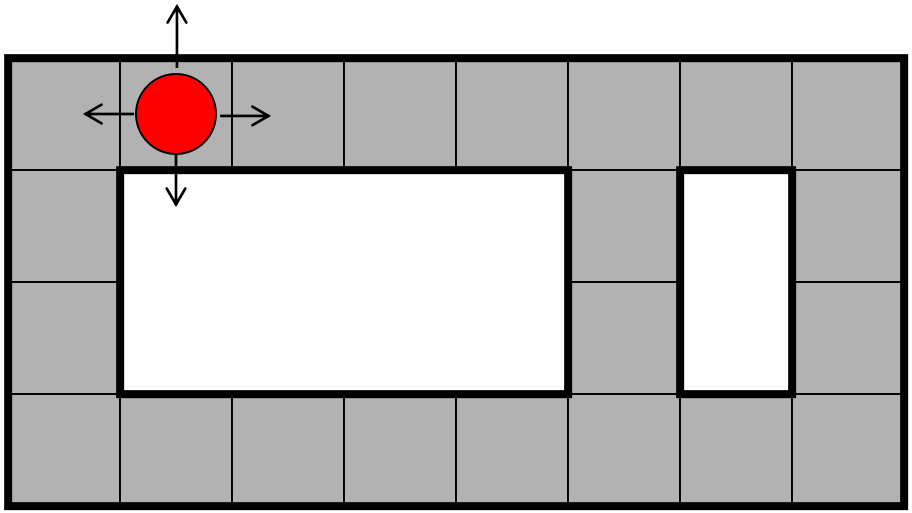
- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options
- Robot tracking:
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)

Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution $B_t(X) = P_t(X_t \mid e_{1,} \ldots e_t)$ (the belief state) over time
- We start with $B_1(X)$ 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

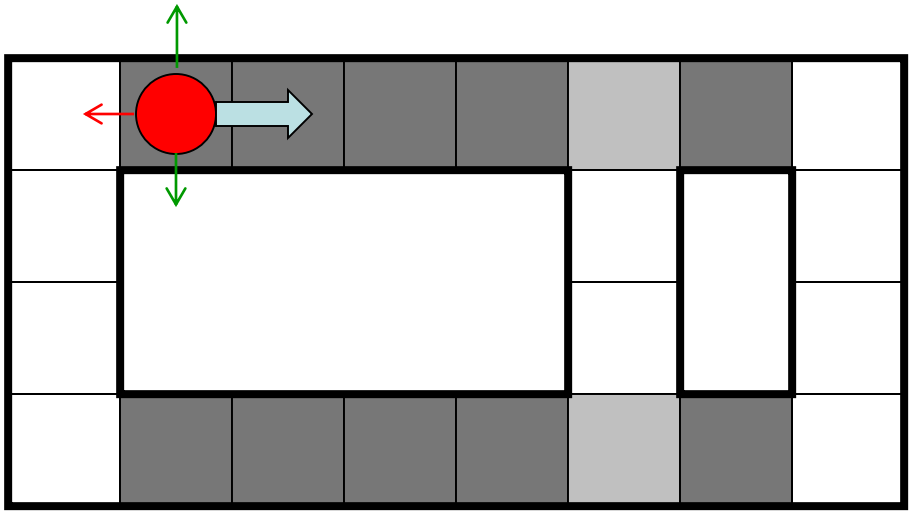


t=0

Sensor model: never more than 1 mistake

Motion model: may not execute action with small prob.

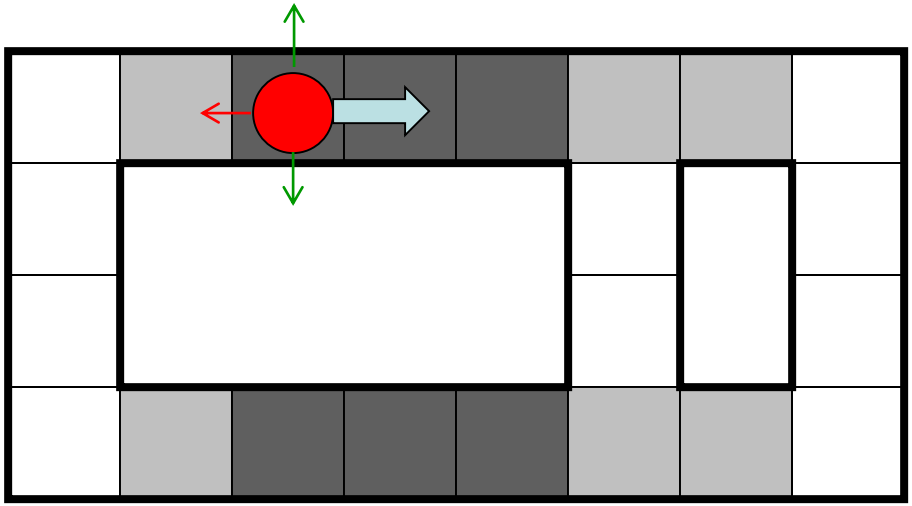
Example: Robot Localization



Prob 0 1

t=1

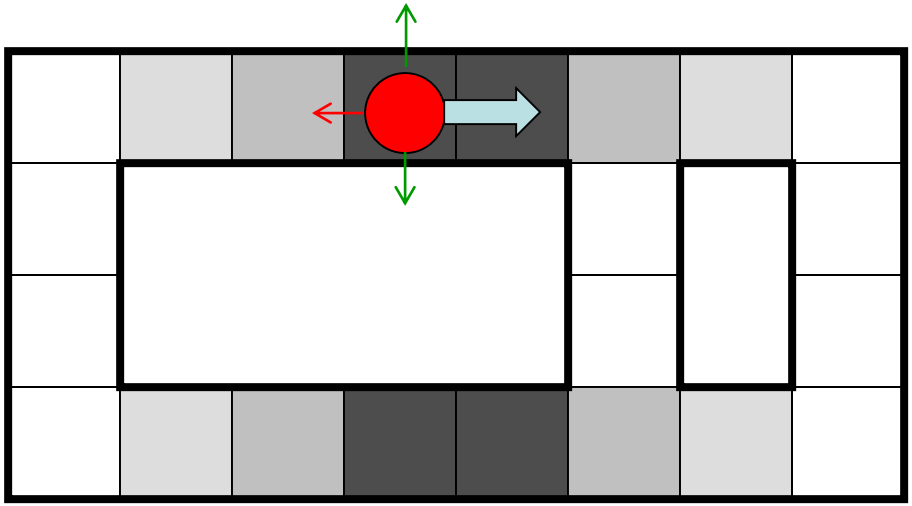
Example: Robot Localization



Prob 0 1

t=2

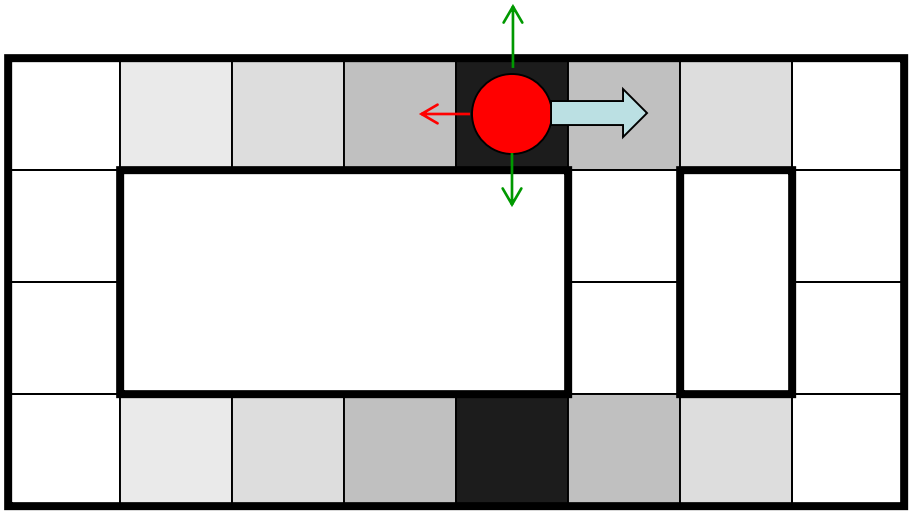
Example: Robot Localization



Prob 0 1

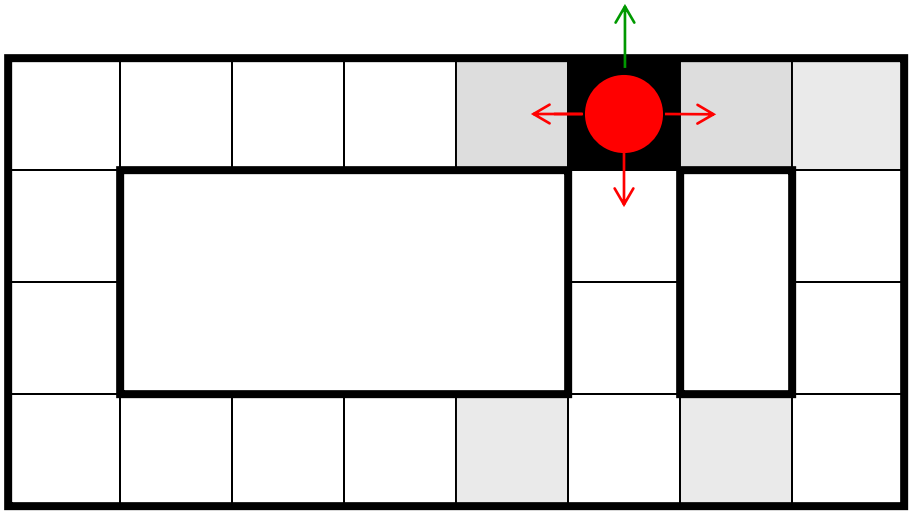
t=3

Example: Robot Localization



t=4

Example: Robot Localization

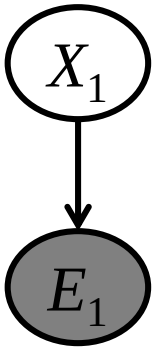


Prob 0 1

t=5

HMM inference: Base cases

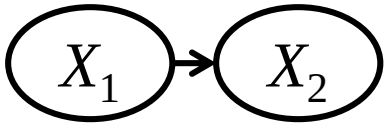
Observation



$$P(X_1|e_1)$$

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

Passage of time



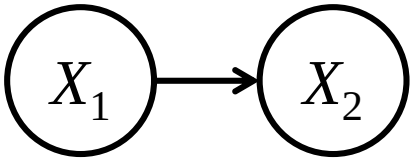
$$P(X_2)$$

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

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:

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

- Or, compactly:

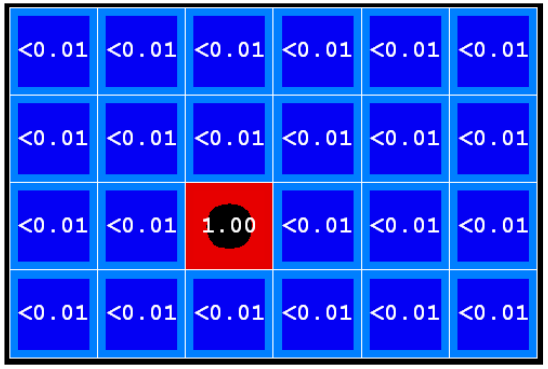
$$B'(X_{t+1}) = \sum_{x_t} P(X'|x)B(x_t)$$

- Basic idea: beliefs get “pushed” through the transitions

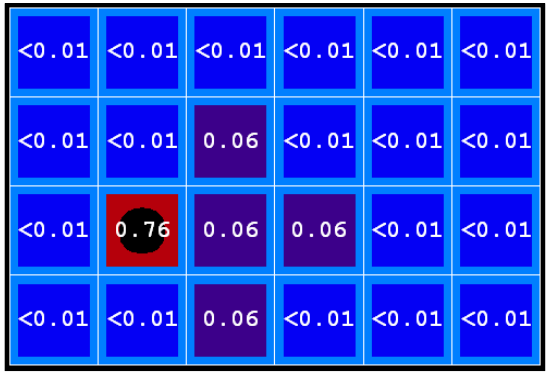
Example: Passage of Time

Transition model: ghosts usually go clockwise

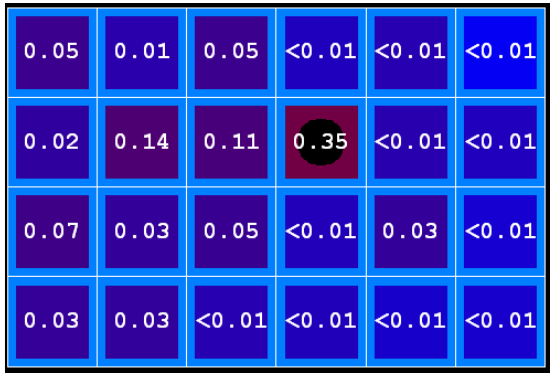
- As time passes, uncertainty “accumulates”



T = 1



T = 2

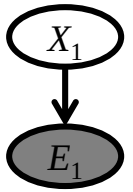


T = 5

Observation

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

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



- Then:

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

- Or:

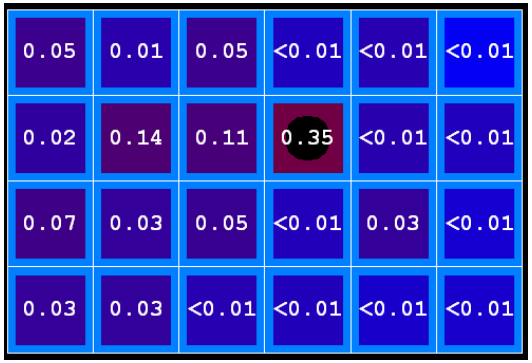
$$B(X_{t+1}) \propto P(e|X)B'(X_{t+1})$$

- Basic idea: beliefs “reweighted” by likelihood of evidence

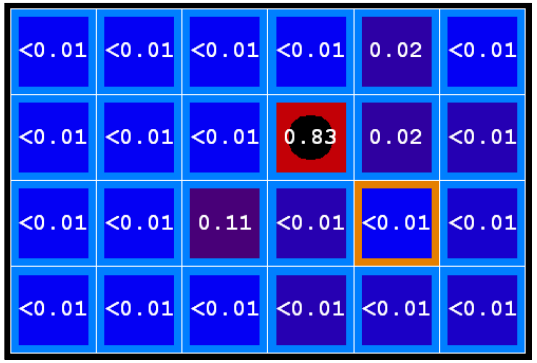
- Unlike passage of time, we have to renormalize

Example: Observation

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



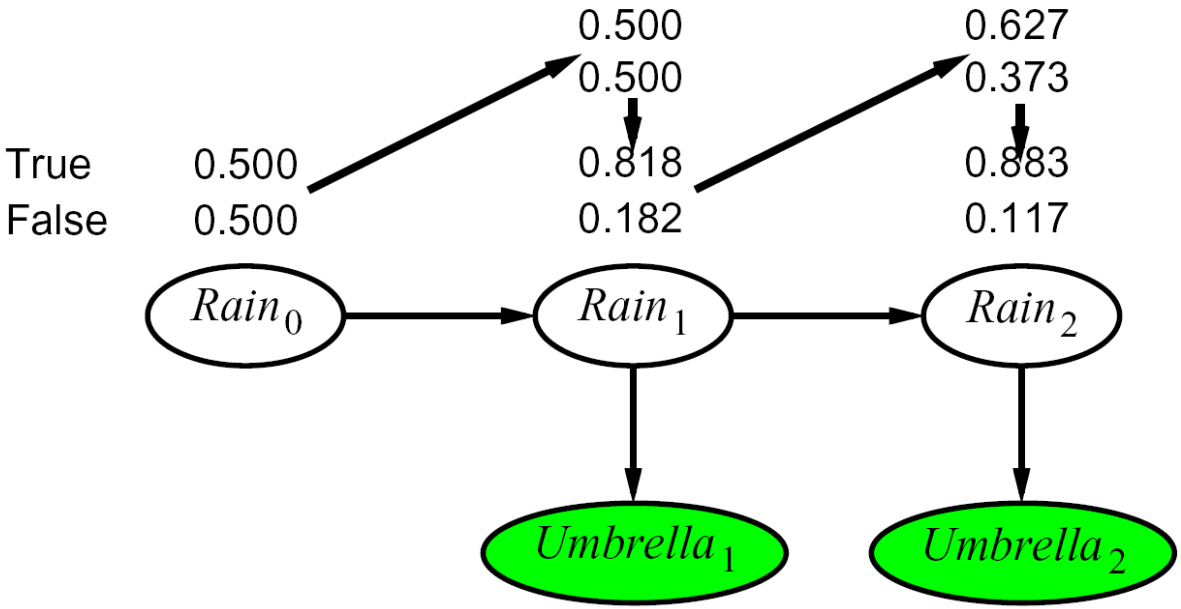
Before observation



After observation

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

Example: Weather HMM



The Forward Algorithm

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

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

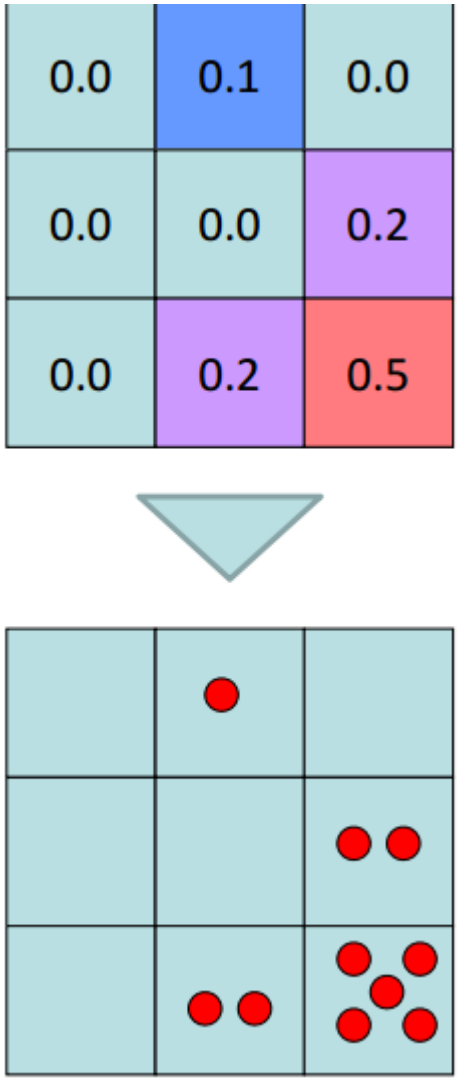
- We can derive the following updates

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

This is exactly variable elimination with order X1, X2, …

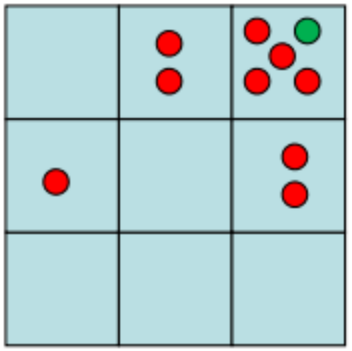
Particle filtering

- 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 may be large
  - In memory: list of particles, not states



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 weight 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

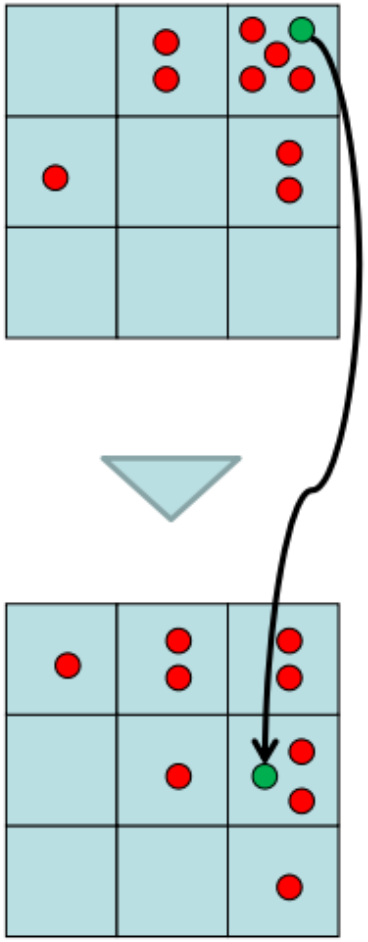
- Each particle is moved by sampling its next position from the transition model

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

  - 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)



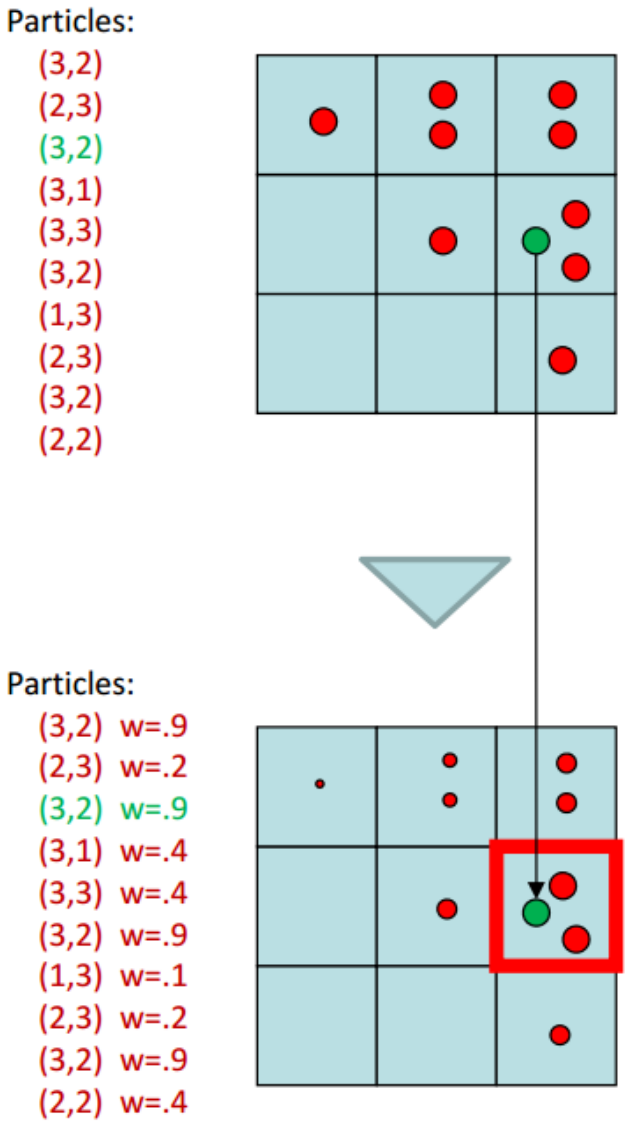
Particle filtering: Observe

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

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

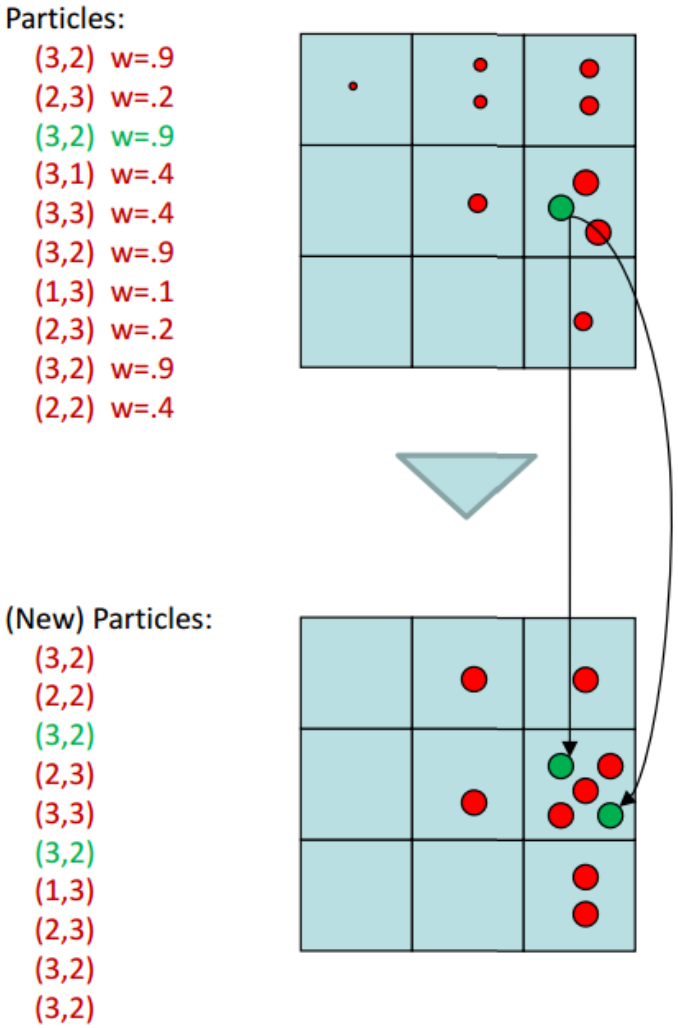
$$B(X) \propto P(e|X)B'(X)$$

  - As before, the probabilities don’t sum to one, since all have been downweighted.



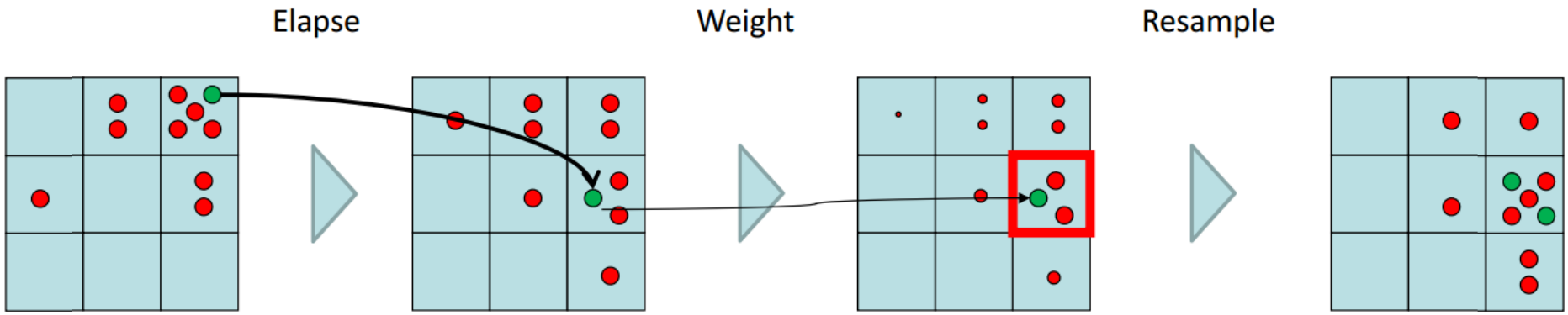
Particle filtering: Resample

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



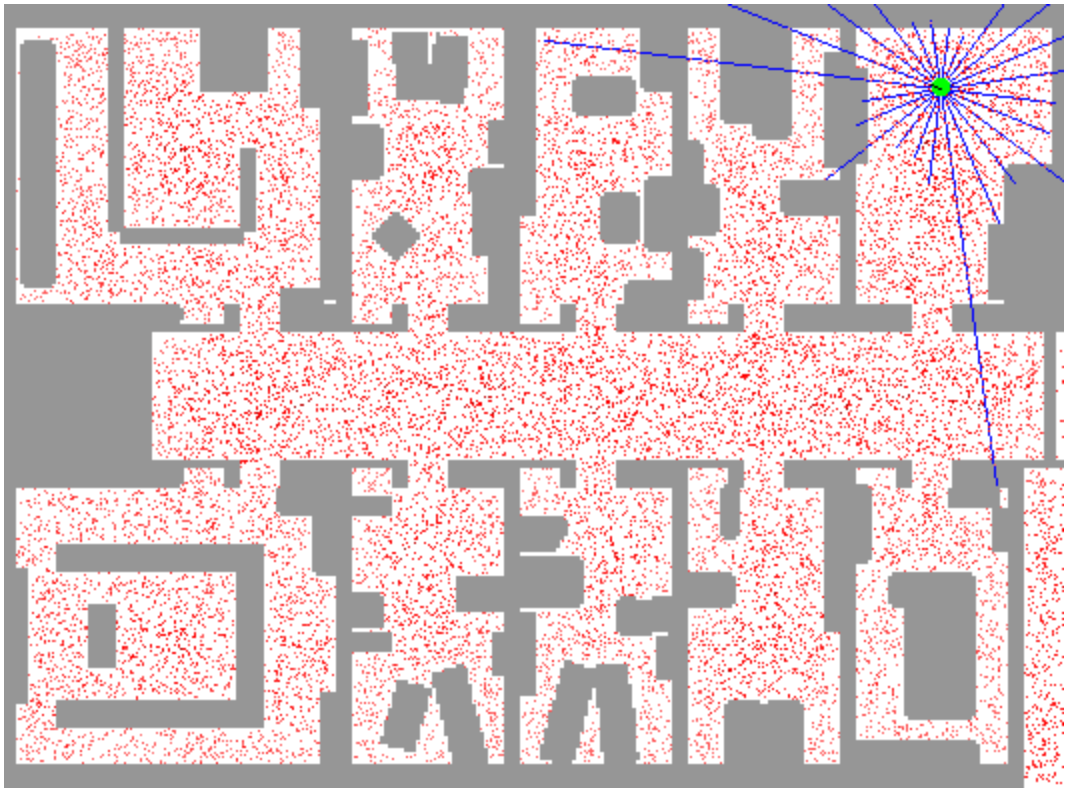
Recap: Particle filtering

- 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)

Example: robot localization



- http://robots.stanford.edu/videos.html

HMMs summary so far

- Markov Models
  - A family of Bayes' nets of a particular regular structure
- Hidden Markov Models (HMMs)
  - Another family of Bayes' nets with regular structure
  - Inference
    - Forward algorithm (repeated variable elimination)
    - Particle filtering (likelihood weighting with some tweaks)