CSE P 573: Artificial Intelligence

Hidden Markov Models

Daniel Weld

University of Washington

[Many of these slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at http://ai.berkeley.edu.]

New Challenge: Partial Observability

- A ghost is in the grid somewhere, but Pacman can't see it!
- Sensor readings tell how close a square is to the ghost
  - On the ghost: red
  - 1 or 2 away: orange
  - 2 or 3 away: yellow
  - 4+ away: green
    - Sensors are ***noisy***, but we know P(Color | Distance)

P(red | 3)	P(orange | 3)	P(yellow | 3)	P(green | 3)
0.05	0.15	0.5	0.3

Etc.

Pacman Maintains a *Belief* about Ghost Locations

Belief = A Probability Distribution over possible locations

Visualized here as color density (each ghost has its own color) Four ghosts shown here

(not to be confused with colors from sensor readings)

Video of Demo Pacman – Sonar (with beliefs)

PROBABILITY REVIEW

5

Random Variables

- A random variable is some aspect of the world about which we (may) have uncertainty
  - R = Is it raining?
  - T = Is it hot or cold?
  - D = How long will it take to drive to work?
  - L = Where is the ghost?
- We denote random variables with Capital Letters
- Random variables have domains
  - T in {hot, cold}
  - D in $[0, \infty)$
  - L in possible locations, maybe {(0,0), (0,1), …}

Joint Distributions

- A *joint distribution* over a set of random variables: $X_1, X_2, \ldots X_n$
  specifies a probability for each assignment (or *outcome*):

$$P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n)$$
$$P(x_1, x_2, \ldots x_n)$$

- Must obey: $P(x_1, x_2, \ldots x_n) \geq 0$

$$\sum_{(x_1, x_2, \ldots x_n)} P(x_1, x_2, \ldots x_n) = 1$$

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- Size of joint distribution if n variables with domain sizes d?
  - For all but the smallest distributions, impractical to write out!

Marginal Distributions

- *Marginal distributions* are **sub-tables** which eliminate variables
- *Marginalization* (summing out): Combine collapsed rows by adding

$P(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(t) = \sum_s P(t, s)$$

$P(T)$

T	P
hot	0.5
cold	0.5

$$P(s) = \sum_t P(t, s)$$

$P(W)$

W	P
sun	0.6
rain	0.4

$$P(X_1 = x_1) = \sum_{x_2} P(X_1 = x_1, X_2 = x_2)$$

Conditional Probabilities

- A simple relation between joint and marginal probabilities
  - In fact, this is taken as the **definition** of a conditional probability

$$P(a|b) = \frac{P(a,b)}{P(b)}$$

$P(T,W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$$P(W=s|T=c) = \frac{P(W=s,T=c)}{P(T=c)} = \boxed{\frac{0.2}{0.5} = 0.4}$$

$$\boxed{\begin{aligned} &= P(W=s,T=c) + P(W=r,T=c) \\ &= 0.2 + 0.3 \ = 0.5 \end{aligned}}$$

Probability Recap

- Conditional probability $P(x|y) = \frac{P(x,y)}{P(y)}$
- Product rule $P(x,y) = P(x|y)P(y)$
- Chain rule
$$\begin{aligned} P(X_1, X_2, \ldots X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1,X_2)\ldots \\ &= \prod_{i=1}^{n} P(X_i|X_1,\ldots,X_{i-1}) \end{aligned}$$
- Bayes rule $P(x|y) = \frac{P(y|x)}{P(y)}P(x)$
- X, Y independent if and only if: $\forall x, y : P(x,y) = P(x)P(y)$
- X and Y are conditionally independent given Z: $X \perp\!\!\!\perp Y | Z$
  if and only if: $\forall x, y, z : P(x,y|z) = P(x|z)P(y|z)$

Probabilistic Inference

- Probabilistic inference =

"compute a desired probability from other known probabilities (e.g. conditional from joint)"

- We generally compute conditional probabilities
  - P(on time | no reported accidents) = 0.90
  - These represent the agent's *beliefs* given the evidence
- Probabilities change with new evidence:
  - P(on time | no accidents, 5 a.m.) = 0.95
  - P(on time | no accidents, 5 a.m., raining) = 0.80
  - Observing new evidence causes *beliefs to be updated*

Outline

- Hidden Markov Models (HMMs)
  - A way to represent a class of probability distribution
- Task of ***Filtering*** (aka Monitoring)
  - HMM Forward Algorithm for Filtering
  - HMM ***Particle Filter*** Representation & Algorithm for Filtering
- Dynamic Bayes Nets
  - An generalization & improvement on HMMs

12

Hidden Markov Models

Cool representation for ***uncertain***, ***sequential*** data

- E.g., ghost locations over time — in Pacman
- E.g., characters on a line — in OCR
- E.g., words over time — in speech recognition

13

Hidden Markov Models

Defines a joint probability distribution:

$$P(X_1, \ldots, X_n, E_1, \ldots, E_n) =$$
$$P(X_{1:n}, E_{1:n}) =$$
$$P(X_1)P(E_1|X_1)\prod_{t=2}^{N} P(X_t|X_{t-1})P(E_t|X_t)$$

Hidden Markov Model: Example

- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t|X_{t-1})$
  - Observations: $P(E|X)$
    Aka "evidence," "emissions"

HMMs have
Stationary transition dynamics
Stationary observation model

Conditional Independence

HMMs have two important independence properties:

- **Future independent of past given the present**

Conditional Independence

HMMs have two important independence properties:

- Future independent of past given the present
- **Current observation independent of all else given current state**

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 observations are ***independent*** given no evidence?
  - [No, correlated by the hidden state]

HMM Computations

Given

- parameters
- evidence $E_{1:n} = e_{1:n}$

Inference problems include:

- Filtering, find $P(X_t|e_{1:t})$ for some t
- Most probable explanation, for some t find

  $x^*_{1:t} = \text{argmax}_{x_{1:t}} P(x_{1:t}|e_{1:t})$
- Smoothing, find $P(X_t|e_{1:n})$ for some $t < n$

Filtering (aka Monitoring)

- **The task of tracking the agent's belief state, B(X), over time**
  - B(X) = distribution over world states; represents agent knowledge
  - We start with B(X) in an initial setting, usually uniform
  - As time passes, or we get observations, we update B(X)
- **Many algorithms for this:**
  - Exact probabilistic inference
  - Particle filter approximation
  - Kalman filter (a method for handling continuous Real-valued random vars)
    - invented in the 60'for Apollo Program – real-valued state, Gaussian noise

Example of HMM Filtering

Robot tracking:

- States (X) are positions on a map (continuous)
- Observations (E) are range readings (continuous)

Example: Robot Localization

Example from Michael Pfeiffer

T=1

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

t=3

Example: Robot Localization

t=4

Example: Robot Localization

t=5

Other Real HMM Examples

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

Ghostbusters HMM

- X = ghost location: x_{11}, … x_{33}
- Ignore pacman location for now

x_{13}	x_{23}	x_{33}
x_{12}	x_{22}	x_{23}
x_{11}	x_{21}	x_{31}

$P(X_1)$

- How specify HMM?

Ghostbusters HMM

- X = ghost locations: x_{11}, … x_{33}
- Ignore pacman location for now – suppose lower left x_{11}

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

Etc…

This is another schema.
Movement noise model is independent of position

- How specify HMM?
  - $P(X_{initial})$ = uniform
  - $P(X_{t+1} \mid X_t)$ =

    A big 9 x 9 table. E.g. $P(X_{t+1} = x_{33} \mid X_t = x_{33})$, …, $P(X_{t+1} = x_{11} \mid X_t = x_{11})$

Ghostbusters HMM

- X = ghost locations: x_{11}, … x_{33}
- Ignore pacman location for now – suppose lower left x_{11}

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

Etc…

- How specify HMM?
  - $P(X_{initial})$ = uniform
  - $P(X_{t+1} \mid X_t)$ = A big 9 x 9 table. E.g. $P(X_{t+1} = x_{33} \mid X_t = x_{33})$, …, $P(X_{t+1} = x_{11} \mid X_t = x_{11})$
  - $P(E_t \mid X_t)$ = also a big table: 4 sonar colors x 9 ghost positions x more if include PM pos

Ghostbusters HMM

- $P(X_1)$ = uniform
- $P(X' \mid X)$ = ghosts usually move clockwise,
  but sometimes move in a random direction or stay put
- $P(E \mid X)$ = same sensor model as before:
  red means probably close, green means likely 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

Etc…

$P(X' \mid X=x_{23})$

P(E|X)

P(red | 3)	P(orange | 3)	P(yellow | 3)	P(green | 3)
0.05	0.15	0.5	0.3

This is ***part*** of a ***schema*** - must specify for other distances and X must include PM position

Filtering (aka Monitoring)

- Filtering, or monitoring, is the task of tracking the distribution B(X) (called "the belief state") over time
- We start with $B_0(X)$ in an initial setting, usually uniform
- We update $B_t(X)$ — *computing $B_{t+1}(X)$*
  1. As time passes, and — *using prob model of how ghosts move*
  2. As we get observations — *using prob model of how noisy sensors work*

Forward Algorithm

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

- t = 0
- $B(X_t)$ = initial distribution
- Repeat forever
  - $B'(X_{t+1})$ = Simulate passage of time from $B(X_t)$
  - Observe e_{t+1}
  - $B(X_{t+1})$ = Update $B'(X_{t+1})$ based on probability of e_{t+1}

36

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:

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

- 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)

T = 1 T = 2 T = 5

Reminder: Forward Algorithm

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

- t = 0
- $B(X_t)$ = initial distribution
- Repeat forever
  - $B'(X_{t+1})$ = Simulate passage of time from $B(X_t)$
  - Observe e_{t+1}
  - $B(X_{t+1})$ = Update $B'(X_{t+1})$ based on probability of e_{t+1}

39

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:

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

$$= P(e_{t+1}|e_{1:t}, X_{t+1})P(X_{t+1}|e_{1:t}) \; /P(e_{t+1}|e_{1:t})$$ *Defn cond prob*

$$= P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t}) \; /P(e_{t+1}|e_{1:t})$$ *Independence*

- Or, compactly:

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

B = belief at t+1 ***after*** observation

B' = belief at t+1 ***before*** observation

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:

$$P(X_{t+1}|e_{1:t+1}) = P(X_{t+1}, e_{t+1}|e_{1:t})/P(e_{t+1}|e_{1:t}) \quad \textit{Defn cond prob}$$

$$= P(e_{t+1}|e_{1:t}, X_{t+1})P(X_{t+1}|e_{1:t}) \; /P(e_{t+1}|e_{1:t}) \quad \textit{Defn cond prob}$$

$$= P(e_{t+1}|X_{t+1})P(X_{t+1}|e_{1:t}) \; /P(e_{t+1}|e_{1:t}) \quad \textit{Independence}$$

- Or, compactly:

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

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

Normalization to Account for Evidence

X	E	P
rain	U	0.4
rain	-	0.1
sun	U	0.2
sun	-	0.3

SELECT the joint probabilities matching the evidence (U in this case)

X	E	P
rain	U	0.4
sun	U	0.2

NORMALIZE the selection (make it sum to one)

$P(W|T = c)$

X	P
rain	0.67
sun	0.33

Since could have seen other evidence, we normalize by dividing by the probability of the evidence we *did* see (in this case dividing by 0.6)…

U = means students have umbrellas

Example: Observation

- As we get observations, beliefs get reweighted, uncertainty "decreases"

Before observation

After observation

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

Quiz: Weather HMM

What is probability of Rain_2 if we've seen umbrellas on both days?

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

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

$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Example: Weather HMM

$B'(x_1=r) = P(x_1=r \mid x_0=r) * 0.5 + P(x_1=r \mid x_0=s) * 0.5$
$= 0.8*0.5 + 0.6*0.5$
$= 0.7$

$B(x_0=r) = 0.5$

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

$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Example: Weather HMM

$B'(x_1=r) = P(x_1=r \mid x_0=r) * 0.5 + P(x_1=r \mid x_0=s) * 0.5$
$= 0.8*0.5 + 0.6*0.5$
$= 0.7$

$B(x_1=r) \propto 0.9 * 0.7 = 0.63$
$B(x_1=s) \propto 0.3 * 0.3 = 0.09$

Divide by 0.72 (=0.63+0.09) to normalize

$B(x_0=r) = 0.5$ $\quad B(x_1=r) = 0.63/0.72 = 0.875$

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

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

$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Example: Weather HMM

$B'(x_1{=}r) = 0.7$

$B'(x_2{=}r) = P(x_2{=}r \mid x_1{=}r)*0.875 + P(x_2{=}r \mid x_1{=}s)*0.125$
$= 0.8*0.875 + 0.6*0.125$
$= 0.775$

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

$B(x_0{=}r) = 0.5$ $B(x_1{=}r) = 0.875$

$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Example: Weather HMM

$B'(x_1{=}r) = 0.7$

$B'(x_2{=}r) = P(x_2{=}r \mid x_1{=}r)*0.875 + P(x_2{=}r \mid x_1{=}s)*0.125$
$= 0.8*0.875 + 0.6*0.125$
$= 0.775$

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

$B(x_1{=}r) \propto 0.9 * 0.775 = 0.6975$
$B(x_1{=}s) \propto 0.3 * 0.225 = 0.0675$
Divide by 0.765 to normalize

$B(x_0{=}r) = 0.5$ $B(x_1{=}r) = 0.875$ $B(x_1{=}r) = 0.912$

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

$P(R_1)$
0.5

R_{t-1}	$P(R_t \mid R_{t-1})$
t	0.8
f	0.6

R_t	$P(U_t \mid R_t)$
t	0.9
f	0.3

Summary: Online Belief Updates

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

1. We update for time:

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

2. 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

(Optimization: don't need to normalize until final time)

Complexity? $O(X^2)$ time

Particle Filtering

Particle Filtering Overview

- ***Approximation technique*** to solve filtering problem
- Represents P distribution with ***samples***
- Filtering still operates in two steps
  - Elapse time
  - Incorporate observations
    - (But this part has two sub-steps: weight & resample)

52

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 exact distribution of 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
- Particle is just new name for ***sample***
- This is how robot localization works in practice

Remember…

We'll start by looking at this

An HMM is defined by:

- Initial distribution: $P(X_1)$
- Transitions: $P(X_t|X_{t-1})$
- Emissions: $P(E|X)$

Here's a Single Particle

- It represents a hypothetical state where the robot is in (1,2)

Particles Approximate Distribution

- Our representation of P(X) is now a list of N particles (samples)
  - Generally, N << |X|

P(x) Distribution

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

P(x=<3,3>) = 5/10 = 50%

Particle Filtering

A more compact view ***overlays*** the samples:

Encodes →

0.0	0.2	0.5
0.1	0.0	0.2
0.0	0.2	0.5

Another Example

In the weather HMM, suppose we decide to approximate the distributions with 5 particles. To initialize the filter, we draw 5 samples from $B(x_0=r) = 0.5$ and we might get the following set of particles:

P(x) Distribution

Particles: S
R
R
S
S

Not such a good approximation, but that's life.

58

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 purpose
- P(x) approximated by **(number of particles with value x) / N**
  - More particles, more accuracy
- What is P((2,2))? 0/10 = 0%
- In fact, many x may have P(x) = 0!

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

Particle Filtering Algorithm

1. Elapse Time
2. Observe
   2a. Downweight samples based on evidence
   2b. Resample

61

Particle Filtering: Elapse Time

- For each particle, x, move x by sampling its next position from the transition model

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

Aka: sample($P(x_{t+1} \mid x_t)$)

  - 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)

Remember: Particle Filtering Algorithm

1. Elapse Time
2. Observe
   - 2a. Downweight samples based on evidence
   - 2b. Resample

63

Particle Filtering: Observe

- Slightly trickier:
  - Don't sample observation, fix it
    - Similar to likelihood weighting,
  - ***For each particle, x, down-weight x based on the evidence***

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

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

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

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 Observe Part II: 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)
(3,1)
(3,2)
(3,1)
(3,2)

Summary: Particle Filtering Algorithm

1. Elapse Time

   Simulate each particle's movement by sampling from $P(x_{t+1} \mid x_t)$

2. Observe

   2a. **Downweight** all samples based on the probability that they would have produced the observed evidence

   2b. **Resample**

Particle Collapse

- Some challenges…
- What if weights of all particles go to zero?
- What if converge to a single particle?

67

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)

[Demos: ghostbusters particle filtering (L15D3,4,5)]

Which Algorithm?

Particle filter, uniform initial beliefs, 300 particles

Which Algorithm?

Exact filter, uniform initial beliefs

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)

[Video: global-sonar-uw-annotated.avi]

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

[Demo: PARTICLES-SLAM-mapping1-new.avi]

Particle Filter SLAM – Video 2

[Demo: PARTICLES-SLAM-fastslam.avi]

Scaling to Large |X|

- 1 Ghost: k (eg 9) possible positions in maze
- 2 Ghosts: k^2 combinations

- N Ghosts: k^N combinations

83

HMM Conditional Independence

- HMMs have two important independence properties:
  - Markov hidden process: future state independent of past given current state
  - Current observation independent of all else given current state

What about Conditional Independence *in Snapshot*

- Can we do something here?
- Factor X into product of (conditionally) independent random vars?

X_3

- Maybe also factor E

E_3

Yes! with Bayes Nets

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