CSE 473: Artificial Intelligence
Spring 2012

Reasoning about Uncertainty
&
Hidden Markov Models

Daniel Weld

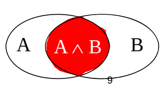
Many slides adapted from Dan Klein, Stuart Russell, Andrew Moore & Luke Zettlemoyer

Outline

- Probabilistic sequence models (and inference)
  - Bayesian Networks – Preview
  - Markov Chains
  - Hidden Markov Models
  - Particle Filters

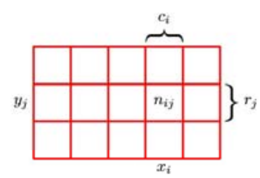
Axioms of Probability Theory

- All probabilities between 0 and 1
  $0 \le P(A) \le 1$
- Probability of truth and falsity
  P(true) = 1 P(false) = 0.
- The probability of disjunction is:
  $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



9

Terminology



Marginal Probability

$p(X = x_i) = \frac{c_i}{N}.$

Joint Probability

$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N}$

Conditional Probability

$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i}$

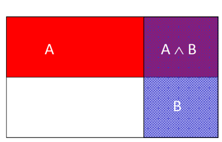
X value is given

Independence

- *A* and *B* are *independent* iff:
  $P(A \mid B) = P(A)$
  $P(B \mid A) = P(B)$
  These constraints logically equivalent
- Therefore, if *A* and *B* are independent:

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)} = P(A)$$

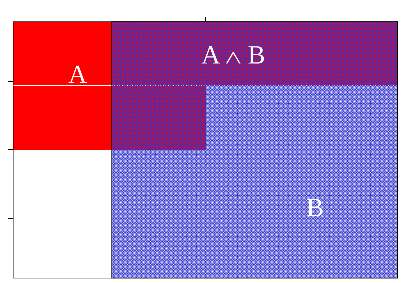
$$P(A \wedge B) = P(A)P(B)$$



14

Conditional Independence

Are A & B independent? P(A|B) ≤ P(A)



P(A)=(.25+.5)/2
= .375

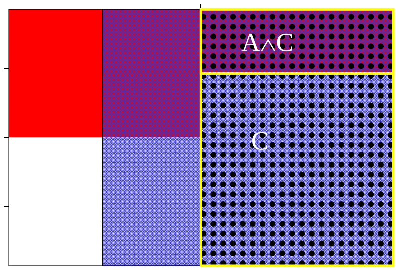
P(B)= .75

P(A|B)=(.25+.25+.5)/3
=.3333

 16

A, B Conditionally Independent Given C

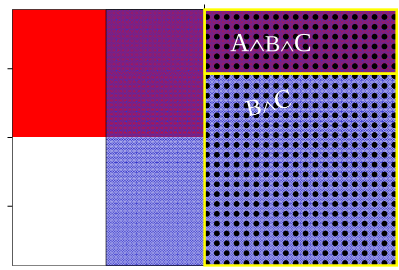
P(A|B,C) = P(A|C)   C = spots



P(A|C) =.25

A, B Conditionally Independent Given C

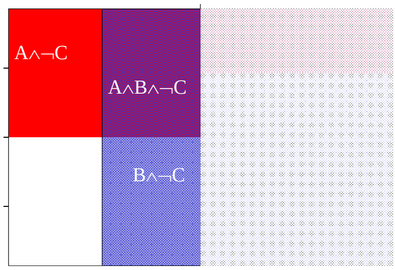
P(A|B,C) = P(A|C)   C = spots



P(A|C) =.25
P(A|B,C)=.25

A, B Conditionally Independent Given ¬C

P(A|B,C) = P(A|C)   C = spots



P(A|C) =.25
P(A|B,C)=.25

P(A|¬C) =.5
P(A|B,¬C)=.5

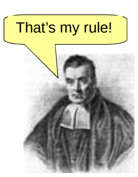
Probabilistic Inference

- *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*

Bayes' Rule

- Two ways to factor a joint distribution over two variables:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$



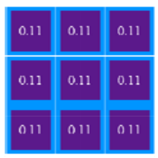
- Dividing, we get:

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

- Why is this at all helpful?
  - Lets us build a conditional from its reverse
  - Often one conditional is tricky but the other one is simple
  - Foundation of many systems we'll see later
- In the running for most important AI equation!

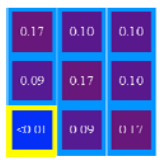
Ghostbusters, Revisited

- Let's say we have two distributions:
  - Prior distribution over ghost location: P(G)
    - Let's say this is uniform
  - Sensor reading model: P(R | G)
    - Given: we know what our sensors do
    - R = reading color measured at (1,1)
    - E.g. P(R = yellow | G=(1,1)) = 0.1



- We can calculate the posterior distribution P(G|r) over ghost locations given a reading using Bayes' rule:

$$P(g|r) \propto P(r|g)P(g)$$



Inference by Enumeration

- General case:
  - Evidence variables: $E_1 \ldots E_k = e_1 \ldots e_k$
  - Query* variable: Q
  - Hidden variables: $H_1 \ldots H_r$

  } $X_1, X_2, \ldots X_n$ *All variables*
- We want: $P(Q|e_1 \ldots e_k)$
- First, select the entries consistent with the evidence
- Second, sum out H to get joint of Query and evidence:

$$P(Q, e_1 \ldots e_k) = \sum_{h_1 \ldots h_r} \underbrace{P(Q, h_1 \ldots h_r, e_1 \ldots e_k)}_{X_1, X_2, \ldots X_n}$$

- Finally, normalize the remaining entries to conditionalize
- Obvious problems:
  - Worst-case time complexity $O(d^n)$
  - Space complexity $O(d^n)$ to store the joint distribution

Bayes' Nets: Big Picture

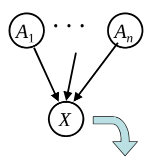
- Two problems with using full joint distribution tables as our probabilistic models:
  - Unless there are only a few variables, the joint is WAY too big to represent explicitly
  - Hard to learn (estimate) anything empirically about more than a few variables at a time
- Bayes' nets: a technique for describing complex joint distributions (models) using simple, local distributions (conditional probabilities)
  - More properly called graphical models
  - We describe how variables locally interact
  - Local interactions chain together to give global, indirect interactions

Bayes' Net Semantics

- Let's formalize the semantics of a Bayes' net
- A set of nodes, one per variable X
- A directed, acyclic graph
- A conditional distribution for each node
  - A collection of distributions over X, one for each combination of parents' values

$$P(X|a_1 \ldots a_n)$$

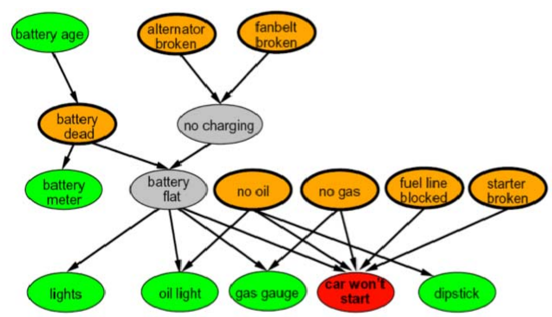
  - CPT: conditional probability table



$$P(X|A_1 \ldots A_n)$$

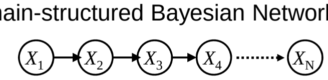
A Bayes net = Topology (graph) + Local Conditional Probabilities

Example Bayes' Net: Car



Markov Models (Markov Chains)

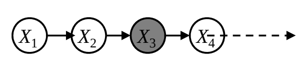
- A Markov model is:
  - a MDP with no actions (and no rewards)
  - a chain-structured Bayesian Network (BN)



- A Markov model includes:
  - Random variables X_t for all time steps t (the state)
  - Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial probs)

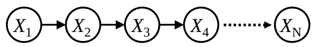
$$P(X_1) \quad \text{and} \quad P(X_t|X_{t-1})$$

Conditional Independence



- Basic conditional independence:
  - Each time step only depends on the previous
  - Future conditionally independent of past given the present
  - This is called the (first order) Markov property
- 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

Markov Models (Markov Chains)



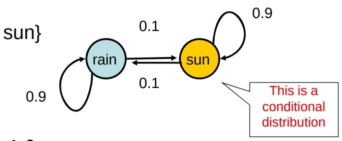
- A Markov model defines
  - a joint probability distribution:

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

- One common inference problem:
  - Compute marginals $P(X_t)$ for some time step, t

Example: Markov Chain

- Weather:
  - States: X = {rain, sun}
  - Transitions:



  - Initial distribution: 1.0 sun
  - What's the probability distribution after one step?

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

$$0.9 \cdot 1.0 + 0.1 \cdot 0.0 = 0.9$$

Markov Chain Inference

- Question: probability of being in state x at time t?
- Slow answer:
  - Enumerate all sequences of length t which end in s
  - Add up their probabilities

$$P(X_t = sun) = \sum_{x_1 \ldots x_{t-1}} P(x_1, \ldots x_{t-1}, sun)$$

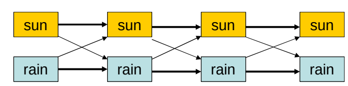
$$P(X_1 = sun)P(X_2 = sun|X_1 = sun)P(X_3 = sun|X_2 = sun)P(X_4 = sun|X_3 = sun)$$

$$P(X_1 = sun)P(X_2 = rain|X_1 = sun)P(X_3 = sun|X_2 = rain)P(X_4 = sun|X_3 = sun)$$

⋮

Mini-Forward Algorithm

- Question: What's P(X) on some day t?
  - We don't need to enumerate all 2^t sequences!



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

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

Forward simulation

Example

- From initial observation of sun

$$\begin{pmatrix}1.0\\0.0\end{pmatrix} \quad \begin{pmatrix}0.9\\0.1\end{pmatrix} \quad \begin{pmatrix}0.82\\0.18\end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix}0.5\\0.5\end{pmatrix}$$

$P(X_1)$ $P(X_2)$ $P(X_3)$ $P(X)$

- From initial observation of rain

$$\begin{pmatrix}0.0\\1.0\end{pmatrix} \quad \begin{pmatrix}0.1\\0.9\end{pmatrix} \quad \begin{pmatrix}0.18\\0.82\end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix}0.5\\0.5\end{pmatrix}$$

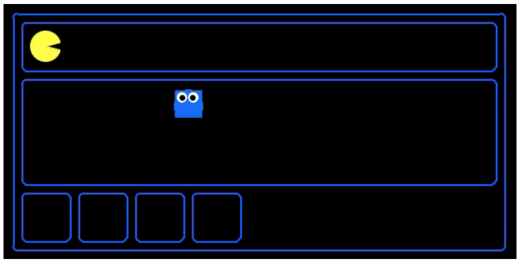
$P(X_1)$ $P(X_2)$ $P(X_3)$ $P(X)$

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

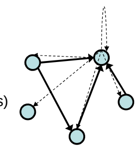
Pac-man Markov Chain

Pac-man knows the ghost's initial position, but gets no observations!



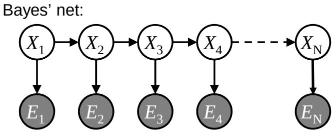
Web Link Analysis

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

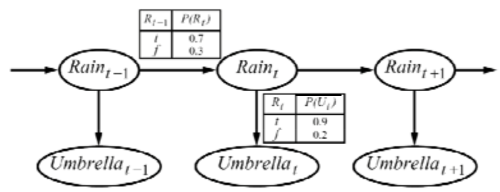


Hidden Markov Models

- 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
  - POMDPs without actions (or rewards).
  - As a Bayes' net:

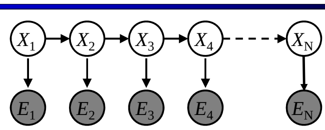


Example



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

Hidden Markov Models



- Defines a joint probability distribution:

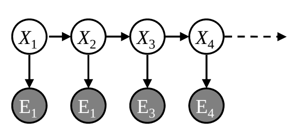
$$P(X_1,\ldots,X_n,E_1,\ldots,E_n) = P(X_{1:n},E_{1:n}) = P(X_1)P(E_1|X_1)\prod_{t=2}^{N} P(X_t|X_{t-1})P(E_t|X_t)$$

Ghostbusters HMM

- $P(X_1)$ = uniform
- P(X'|X) = usually move clockwise, but sometimes move in a random direction or stay in place
- P(E|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>)

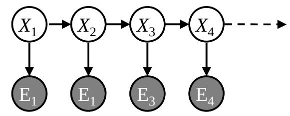
P(E|X)	P(red | 3)	P(orange | 3)	P(yellow | 3)	P(green | 3)
	0.05	0.15	0.5	0.3

HMM Computations

- Given
  - joint $P(X_{1:n}, E_{1:n})$
  - evidence $E_{1:n} = e_{1:n}$
- Inference problems include:
  - Filtering, find $P(X_t|e_{1:t})$ for all t
  - Smoothing, find $P(X_t|e_{1:n})$ for all t
  - Most probable explanation, find
    $x^*_{1:n} = \text{argmax}_{x_{1:n}} P(x_{1:n}|e_{1:n})$

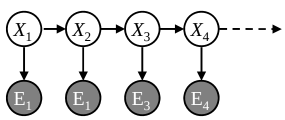
Real HMM Examples

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



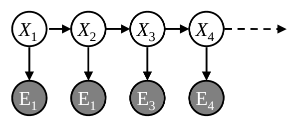
Real HMM Examples

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



Real HMM Examples

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

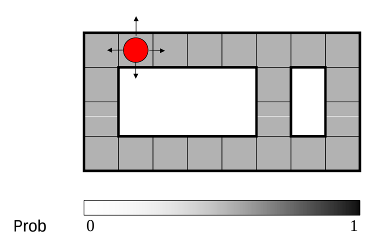


Filtering / Monitoring

- Filtering, or monitoring, is the task of tracking the distribution B(X) (the belief state) over time
- We start with B(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

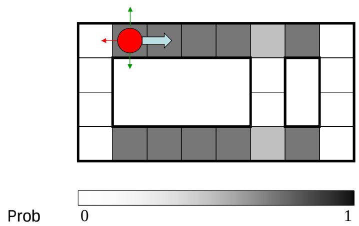


t=0

Sensor model: never more than 1 mistake

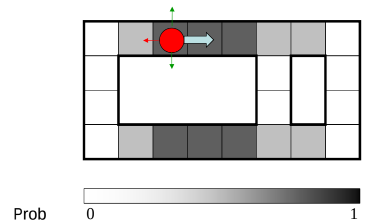
Motion model: may not execute action with small prob.

Example: Robot Localization



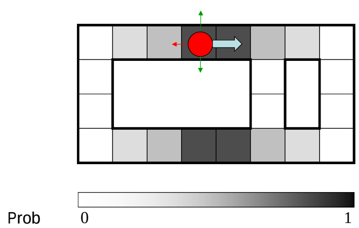
t=1

Example: Robot Localization



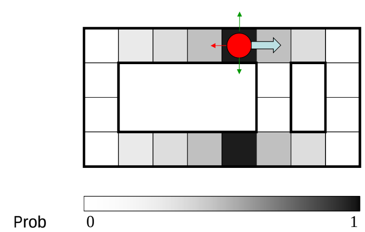
t=2

Example: Robot Localization



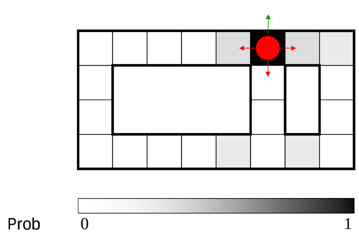
t=3

Example: Robot Localization



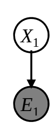
t=4

Example: Robot Localization



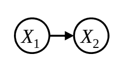
t=5

Inference Recap: Simple Cases



$P(X_1|e_1)$

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

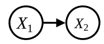


$P(X_2)$

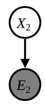
$$P(x_2) = \sum_{x_1} P(x_1, x_2)$$
$$= \sum_{x_1} P(x_1)P(x_2|x_1)$$

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:

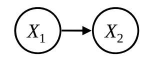
$$P(x_t|e_{1:t}) \propto_X P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$

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:



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

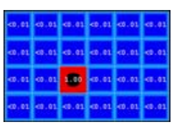
- Or, compactly:

$$B'(X') = \sum_x P(X'|x)B(x)$$

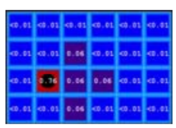
- Basic idea: beliefs get "pushed" through the transitions
  - With the "B" notation, we have to be careful about what time step t the belief is about, and what evidence it includes

Example: Passage of Time

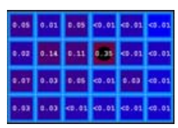
- As time passes, uncertainty "accumulates"



T = 1



T = 2



T = 5

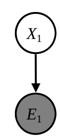
$$B'(X') = \sum_x P(X'|x)B(x)$$

Transition model: ghosts usually go clockwise

Observation

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

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



- Then:

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

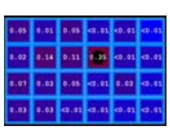
- Or:

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

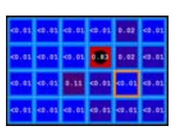
- Basic idea: beliefs reweighted by likelihood of evidence
- Unlike passage of time, we have to renormalize

Example: Observation

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



Before observation



After observation

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

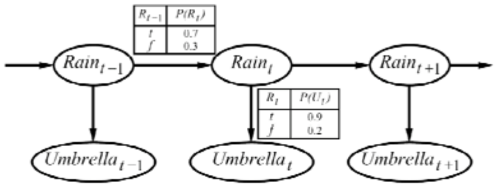
The Forward Algorithm

- We want to know: $B_t(X) = P(X_t|e_{1:t})$
- We can derive the following updates

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

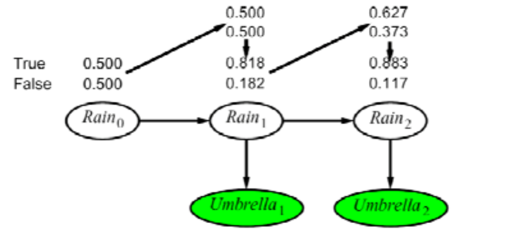
- To get $B_t(X)$ compute each entry and normalize

Example: Run the Filter

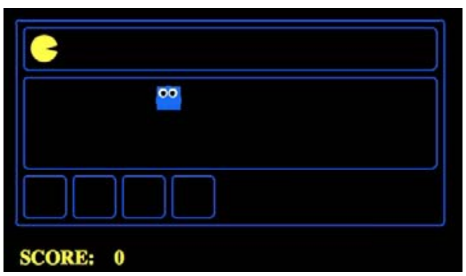


- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t|X_{t-1})$
  - Emissions: $P(E|X)$

Example HMM



Example Pac-man



Summary: Filtering

- Filtering is the inference process of finding a distribution over X_T given e_1 through e_T : $P(X_T | e_{1:t})$
- We first compute $P(X_1 | e_1)$: $P(x_1|e_1) \propto P(x_1) \cdot P(e_1|x_1)$
- For each t from 2 to T, we have $P(X_{t-1} | e_{1:t-1})$
- **Elapse time:** compute $P(X_t | e_{1:t-1})$

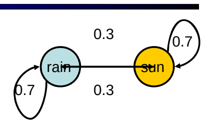
$$P(x_t|e_{1:t-1}) = \sum_{x_{t-1}} P(x_{t-1}|e_{1:t-1}) \cdot P(x_t|x_{t-1})$$

- **Observe:** compute $P(X_t | e_{1:t-1} , e_t) = P(X_t | e_{1:t})$

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

Recap: Reasoning Over Time

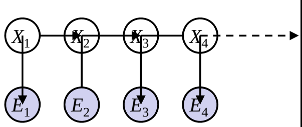
- Stationary Markov models



$P(X_1)$ $P(X|X_{-1})$

$P(E|X)$

- Hidden Markov models



X	E	P
rain	umbrella	0.9
rain	no umbrella	0.1
sun	umbrella	0.2
sun	no umbrella	0.8

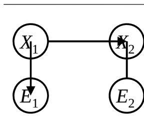
Recap: Filtering

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

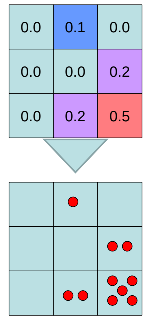
$$P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)$$



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

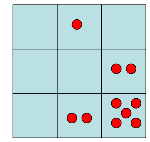
Particle Filtering

- Sometimes |X| is too big to use exact inference
  - |X| may be too big to even store B(X)
  - E.g. X is continuous
  - $|X|^2$ may be too big to do updates
- 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



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 will have P(x) = 0!
  - More particles, more accuracy
- For now, all particles have a weight of 1



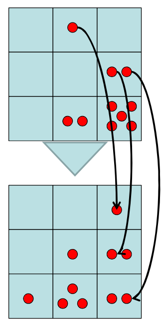
Particles:
(3,3)
(2,3)
(3,3)
(3,2)
(3,3)
(3,2)
(2,1)
(3,3)
(3,3)
(2,1)

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 probs
  - 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)



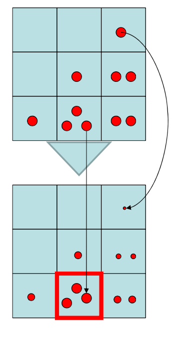
Particle Filtering: Observe

- Slightly trickier:
  - Don't do rejection sampling (why not?)
  - We don't sample the observation, we fix it
  - This is similar to likelihood weighting, so we downweight our samples based on the evidence

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

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

  - Note that, as before, the probabilities don't sum to one, since most have been downweighted (in fact they sum to an approximation of P(e))

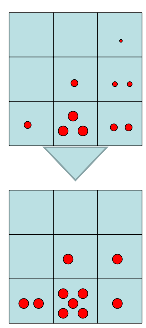


Particle Filtering: Resample

- 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



Particle Filtering Summary

- Represent current belief P(X | evidence to date) as set of *n* samples (actual assignments X=x)
- For each new observation e:
  1. Sample transition, once for each current particle x

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

  2. For each new sample x', compute importance weights for the new evidence e:

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

  3. Finally, normalize the importance weights and resample N new particles

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

Robot Localization

QuickTime™ and a
GIF decompressor
are needed to see this picture.

Which Algorithm?

Exact filter, uniform initial beliefs

Which Algorithm?

Particle filter, uniform initial beliefs, 300 particles

Which Algorithm?

Particle filter, uniform initial beliefs, 25 particles