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
  - Exact Inference
  - Particle Filters

Going Hunting

Inference by Enumeration

- General case:
  - Evidence variables: $E_1 \ldots E_k = e_1 \ldots e_k$
  - Query variable: $Q$
  - Hidden variables: $H_1 \ldots H_r$
  - All variables $X_1, X_2, \ldots, X_n$

- 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:
  $$P(Q, e_1 \ldots e_k) = \sum_{h_1 \ldots h_r} P(Q, h_1 \ldots h_r, e_1 \ldots e_k)$$
- Finally, normalize the remaining entries to conditionals

- 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

Formally:

- A set of nodes, one per variable $X$
- A directed, acyclic graph
- A CPT for each node
  - CPT is "Conditional Probability Table"
  - Collection of distributions over $X$, one for each combination of parents’ values
    $$P(X | a_1 \ldots a_n)$$

A Bayes net = Topology (graph) + Local Conditional Probabilities
**Example Bayes’ Net: Car**

- Battery age
- Alarms
- Fanbelt Check
- Battery charged
- Lights
- Oil
- Gas
- Far enough
  - Switch

**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:

**Hidden Markov Models**

- Defines a joint probability distribution:
  - $P(X_1, \ldots, X_n, E_1, \ldots, E_n) = P(X_1, E_1) \prod_{i=2}^{n} P(X_i | X_{i-1}) P(E_i | X_i)$

**Example**

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

**Ghostbusters HMM**

- $P(X_t) = \text{uniform}$
- $P(X_t | X_{t-1}) = \text{usually move clockwise, but sometimes}
  \text{move in a random direction or stay in place}$
- $P(E_t | X_t) = \text{same sensor model as before}$
  - red means close, green means far away.

**HMM Computations**

- Given
  - joint $P(X_t, E_{1:t})$
  - evidence $E_{1:n} = e_{1:n}$

- Inference problems include:
  - Filtering, find $P(X_t | e_{1:t})$ for some $t$
  - Smoothing, find $P(X_t | e_{1:n})$ for some $t$
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 \( x^*_{1:n} = \arg \max_{x_{1:n}} P(x_{1:n}|e_1:n) \)

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

- **Sensor model**: never more than 1 mistake
- **Motion model**: may not execute action with small prob.

\[ \text{Prob} \]
\[ t=0 \]
\[ \text{Sensor model: never more than 1 mistake} \]
\[ \text{Motion model: may not execute action with small prob.} \]

Example from Michael Pfeiffer
Inference Recap: Simple Cases

That's my rule!

Online Belief Updates

- Every time step, we start with current $P(X \mid \text{evidence})$
- We update for time:
  
  $$P(x_t|e_{t-1}) = \sum_{e_{t-1}} P(x_{t-1}|e_{t-1}) \cdot P(x_t|x_{t-1})$$

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

Example: Passage of Time

Without observations, uncertainty "accumulates"

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

Transition model: ghosts usually go clockwise

Passage of Time

- Assume we have current belief $P(X \mid \text{evidence to date})$
  
  $$B(X_t) = P(X_t|e_{1:t})$$

- 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

Observations

- Assume we have current belief $P(X \mid \text{previous evidence})$:
  
  $$B'(X_{t+1}) = P(X_{t+1}|e_{1:t})$$

- 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

\[
P(x_t|e_{1:t}) \propto \frac{P(x_t|e_{1:t})}{\sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t})}
= \frac{\sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}, x_t) P(x_t|e_t|x_{t-1})}{P(e_t|x_t) \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}, x_t) P(x_t|e_t|x_{t-1})}
= P(e_t|x_t) \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}, x_t) P(x_t|e_t|x_{t-1})
\]

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

Example

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

Forward Algorithm

Summary: Filtering

- Filtering is the inference process of finding a distribution over \( X_t \) given \( e_t \) 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 \( P(X_t|e_{1:t}) \)
- Elapse time: compute \( P(X_t|e_{1:a}) \)

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

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

\[
P(x_t|e_{1:t}) \propto P(x_t|e_{1:t-1}) \cdot P(e_t|x_t)
\]
Recap: Reasoning Over Time

- **Stationary Markov models**
  \[
  P(X_t) \quad P(X|X_{t-1})
  \]

- **Hidden Markov models**
  \[
  X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow \cdots
  \]

**Particle Filtering**

- Sometimes |X| is too big for exact inference
  - |X| may be too big to even store B(X)
  - E.g. when 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
  - How robot localization works in practice

**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

**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**

- How handle noisy observations?
- Suppose sensor gives red reading?
### 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.

<table>
<thead>
<tr>
<th>Old Particles:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,3) w=0.1</td>
</tr>
<tr>
<td>(2,1) w=0.9</td>
</tr>
<tr>
<td>(3,1) w=0.4</td>
</tr>
<tr>
<td>(2,2) w=0.3</td>
</tr>
<tr>
<td>(2,1) w=0.9</td>
</tr>
<tr>
<td>(3,2) w=0.3</td>
</tr>
<tr>
<td>(1,1) w=0.4</td>
</tr>
<tr>
<td>(2,1) w=0.9</td>
</tr>
<tr>
<td>(2,1) w=0.9</td>
</tr>
<tr>
<td>(3,1) w=0.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>New Particles:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2,1) w=1</td>
</tr>
<tr>
<td>(2,1) w=1</td>
</tr>
<tr>
<td>(2,1) w=1</td>
</tr>
<tr>
<td>(3,2) w=1</td>
</tr>
<tr>
<td>(2,2) w=1</td>
</tr>
<tr>
<td>(2,1) w=1</td>
</tr>
<tr>
<td>(1,1) w=1</td>
</tr>
<tr>
<td>(3,1) w=1</td>
</tr>
<tr>
<td>(2,1) w=1</td>
</tr>
<tr>
<td>(1,1) w=1</td>
</tr>
</tbody>
</table>

### Particle Filtering Summary

- Represent current belief $P(X \mid \text{evidence to date})$ as set of $n$ samples (actual assignments $X=x$).
- For each new observation $e$:
  1. Sample transition, once for each current particle $x$.
  \[ x' = \text{sample}(P(X'=x \mid e)) \]
  2. For each new sample $x'$, compute importance weights for the new evidence $e$:
  \[ w(x') = P(e \mid x') \]
  3. Finally, normalize by resampling the importance weights to create $N$ 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.

### Which Algorithm?

- Particle filter, uniform initial beliefs, 25 particles.
- Particle filter, uniform initial beliefs, 300 particles.
Which Algorithm?

Exact filter, uniform initial beliefs

P4: Ghostbusters

- **Plot**: Pacman’s grandfather, Grandpac, learned to hunt ghosts for sport.

- **He was blinded by his power, but could hear the ghosts’ banging and clanging.**

- **Transition Model**: All ghosts move randomly, but are sometimes biased

- **Emission Model**: Pacman knows a “noisy” distance to each ghost

<table>
<thead>
<tr>
<th>Noisy distance prob</th>
<th>True distance = 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>