Dynamic Bayesian Networks

CSE 473
473 Topics

<table>
<thead>
<tr>
<th>Inference</th>
<th>Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic</td>
<td>Knowledge Representation</td>
</tr>
<tr>
<td></td>
<td>Planning</td>
</tr>
<tr>
<td></td>
<td>Probability</td>
</tr>
<tr>
<td></td>
<td>Reinforcement Learning</td>
</tr>
</tbody>
</table>

© Daniel S. Weld
Last Time

• Basic notions
• Bayesian networks
• Statistical learning
  Parameter learning (MAP, ML, Bayesian L)
  Naïve Bayes
  Structure Learning
  Expectation Maximization (EM)
• Dynamic Bayesian networks (DBNs)
• Markov decision processes (MDPs)
Generative Planning

Input
- Description of (initial state of) world \textit{(in some KR)}
- Description of goal \textit{(in some KR)}
- Description of available actions \textit{(in some KR)}

Output
- Controller
  - E.g. Sequence of actions
  - E.g. Plan with loops and conditionals
  - E.g. Policy = $f: \text{states} \rightarrow \text{actions}$
Simplifying Assumptions

Environment

Static vs. Dynamic

Instantaneous vs. Durative

Deterministic vs. Stochastic

Perceps

Fully Observable vs. Partially Observable

Perfect vs. Noisy

Actions

What action next?

Full vs. Partial satisfaction
Static
Deterministic
Observable
Instantaneous
Propositional

“Classical Planning”
Uncertainty

- **Disjunctive**
  
  Next state could be one of a set of states.

- **Probabilistic**
  
  Next state is a probability distribution over the set of states.

*Which is a more general model?*
STRIPS Action Schemata

• Instead of defining ground actions:  
  pickup-A and pickup-B and ...
• Define a schema:

\[
(:\text{operator} \text{ pick-up} \\
  :\text{parameters } ((\text{block } ?\text{ob1})) \\
  :\text{precondition} (\text{and} (\text{clear } ?\text{ob1}) \\
  (\text{on-table } ?\text{ob1}) \\
  (\text{arm-empty})) \\
  :\text{effect} (\text{and} (\text{not} (\text{clear } ?\text{ob1})) \\
  (\text{not} (\text{on-table } ?\text{ob1})) \\
  (\text{not} (\text{arm-empty})) \\
  (\text{holding } ?\text{ob1})))
\]

Note: strips doesn’t allow derived effects; you must be complete!
Nondeterministic “STRIPS”?

pick_up_A

- arm_empty
- on_table_A
- holding_A

arm_empty

clear_A

? 

on_table_A

? 

+ arm_empty
Probabilistic STRIPS?

pick_up_A

arm_empty

clear_A

on_table_A

P<0.9

+ arm_empty
- on_table_A
- holding_A

But what does this all mean anyway?
Defn: Markov Model

Q: set of states

\( \pi \): init prob distribution

A: transition probability distribution
Probability Distribution, A

• **Forward Causality**
  The probability of $s^t$ does not depend *directly* on values of future states.

• **Probability of new state could depend on**
  The history of states visited.
  \[ \Pr(s^t|s^{t-1},s^{t-2},...,s^0) \]

• **Markovian Assumption**
  \[ \Pr(s^t|s^{t-1},s^{t-2},...,s^0) = \Pr(s^t|s^{t-1}) \]

• **Stationary Model Assumption**
  \[ \Pr(s^t|s^{t-1}) = \Pr(s^k|s^{k-1}) \text{ for all } k. \]
A general stochastic process (a), a Markov chain (b), and a stationary Markov chain (c).

What is the cost of representing a stationary probability distribution?

Can we do something better?
Representing A

Q: set of states
π: init prob distribution
A: transition probabilities
how can we represent these?

Probability of transitioning from $s_1$ to $s_2$
Factoring Q

- Represent Q simply as a set of states?
- Is their internal structure?
  Consider a robot domain
  What is the state space?
A Factored domain

• Variables:
  has_user_coffee (huc),
  has_robot_coffee (hrc),
  robot_is_wet (w),
  has_robot_umbrella (u),
  raining (r),
  robot_in_office (o)

• How many states?

\[ 2^6 = 64 \]
Representing \( \pi \) Compactly

Q: set of states
\( \pi \): init prob distribution

How represent this efficiently?

With a Bayes net (of course!)
Representing A Compactly

Q: set of states
π: init prob distribution
A: transition probabilities

How big is matrix version of A? 4096
Dynamic Bayesian Network

Total values required to represent transition probability table = 36

Vs. 4096 required for a complete state probability table?
Dynamic Bayesian Network

Defined formally as
* Set of random variables
* BN for initial state
* 2-layer DBN for transition

Also known as a
Factored Markov Model
This is Really a “Schema”

“Unrolling”
Semantics:
A DBN $\rightarrow$ A Normal Bayesian Network
Multiple Actions

- **Variables**: 
  
  has_user_coffee (huc), has_robot_coffee (hrc), robot_is_wet (w), has_robot_umbrella (u), raining (r), robot_in_office (o)

- **Actions**: 
  
  buy_coffee, deliver_coffee, get_umbrella, move

- **We need an extra variable**
- **(Alternatively a separate 2DBN / action)**
Actions in DBN
What can we do with DBNs?

- State estimation
- Planning
  - If you add rewards
  - (Next class)
State Estimation

OR

Huc
Hrc

U
W

Move

? → BuyC

... → ?
Noisy Observations

True state vars (unobserved)

Noisy, sensor reading

Action var known with certainty

\[ T \quad T+1 \]

\[ \text{So} \]

\[ \text{huc} \quad \text{hrc} \quad \text{w} \quad \text{u} \quad \text{r} \quad \text{o} \]

\[ \begin{array}{c}
\text{a} \\
\end{array} \]

\[ \begin{array}{c}
\text{So} \\
\end{array} \]

\[ \begin{array}{cc}
0 & 0 \\
.9 & .2 \\
\end{array} \]
State Estimation

OR
Huc
Hrc
UW

Move

?  BuyC  ...

So=T

© Daniel S. Weld
Maybe we don’t know action!

True state vars (unobserved)

Noisy, sensor reading

© Daniel S. Weld
State Estimation

O R Huc Hrc U W

So = T ...

© Daniel S. Weld
Inference

- Expand DBN (schema) into BN
  Use inference as shown last week
  Problem?
- Particle Filters
Particle Filters

- **Create 1000s of “Particles”**
  Each one has a “copy” of each random var
  These have concrete values
  Distribution is approximated by whole set
  - Approximate technique! Fast!

- **Simulate**
  Each time step, iterate per particle
  - Use the 2 layer DBN + random # generator to
  - Predict next value for each random var
  Resample!
Resampling

• **Simulate**
  Each time step, iterate per particle
  • Use the 2 layer DBN + random # generator to
  • Predict next value for each random var

**Resample:**
• Given observations (if any) at new time point
• Calculate the likelihood of each particle
• Create a new population of particles by
  - Sampling from old population
  - Proportionally to likelihood