Simba Technologies Tech Talk/Info Session
Mon., Sept 21
6 – 7 pm
DMP 310

EA Info Session
Tues., Sept 22
6 – 7 pm
DMP 310

Co-op Drop-in FAQ Session
Thurs., Sept 24
12:30 – 1:30 pm
Reboot Cafe

Resume Editing Drop-in Sessions
Mon., Sept 28
10 am – 2 pm (sign up at 9 am)
ICCS 253

Facebook Crush Your Code Workshop
Mon., Sept 28
6 – 8 pm
DMP 310

UBC Careers Day & Professional School Fair
Wed., Sept 30 & Thurs., Oct 1
10 am – 3 pm
AMS Nest
Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 6

Sep, 21, 2015

Slide credit POMDP: C. Conati and P. Viswanathan
Lecture Overview

Partially Observable Markov Decision Processes

• Summary
  • Belief State
  • Belief State Update

• Policies and Optimal Policy
Markov Models

- Markov Chains
- Hidden Markov Model
- Partially Observable Markov Decision Processes (POMDPs)
- Markov Decision Processes (MDPs)

Noisy Observations
Noisy Actions
Rewards
Belief State and its Update

\[ b(s) \]

\[ b'(s') = \alpha P(e \mid s') \sum_s P(s' \mid s, a) b(s) \]

as

\[ b' = \text{Forward}(b, a, e) \]

➢ To summarize: when the agent performs action \( a \) in belief state \( b \), and then receives observation \( e \), filtering gives a unique new probability distribution over state

- \textit{deterministic transition from one belief state to another}
Optimal Policies in POMDs?

Theorem (Astrom, 1965):

- The optimal policy in a POMDP is a function $\pi^*(b)$ where $b$ is the belief state (probability distribution over states).

That is, $\pi^*(b)$ is a function from belief states (probability distributions) to actions:

- It does not depend on the actual state the agent is in.
- Good, because the agent does not know that, all it knows are its beliefs!

Decision Cycle for a POMDP agent:

- Given current belief state $b$, execute $a = \pi^*(b)$.
- Receive observation $e$.
- Compute: $b'(s') = \alpha P(e | s') \sum_s P(s' | s, a) b(s)$.
- Repeat.
How to Find an Optimal Policy?

- Turn a POMDP into a corresponding MDP and then solve that MDP
- Generalize VI to work on POMDPs
- Develop Approx. Methods
  - Point-Based VI
  - Look Ahead
Finding the Optimal Policy: State of the Art

- Turn a POMDP into a corresponding MDP and then apply VI: only small models

- Generalize VI to work on POMDPs
  - 10 states in 1998
  - 200,000 states in 2008-09

- Develop Approx. Methods
  - Point-Based VI and Look Ahead
  - Even 50,000,000 states
    http://www.cs.uwaterloo.ca/~ppoupart/software.html
Dynamic Decision Networks (DDN)

- Comprehensive approach to agent design in partially observable, stochastic environments

- Basic elements of the approach
  - Transition and observation models are represented via a Dynamic Bayesian Network (DBN).
  - The network is extended with decision and utility nodes, as done in decision networks.
Dynamic Decision Networks (DDN)

• A filtering algorithm is used to incorporate each new percept and the action to update the belief state $X_t$

• Decisions are made by projecting forward possible action sequences and choosing the best one: *look ahead search*
Dynamic Decision Networks (DDN)

- Nodes in yellow are known (evidence collected, decisions made, local rewards)
- Agent needs to make a decision at time $t$ ($A_t$ node)
- Network unrolled into the future for 3 steps
- Node $U_{t+3}$ represents the utility (or expected optimal reward $V^*$) in state $X_{t+3}$
  - i.e., the reward in that state and all subsequent rewards
  - Available only in approximate form (from another approx. method)
Look Ahead Search for Optimal Policy

General Idea:

- **Expand the decision process for n steps into the future, that is**
  - “Try” all actions at every decision point
  - Assume receiving all possible observations at observation points

- **Result: tree of depth 2n+1** where
  - every branch represents one of the possible sequences of n actions and n observations available to the agent, and the corresponding belief states
  - The leaf at the end of each branch corresponds to the *belief state* reachable via that sequence of actions and observations – use filtering to compute it

- “**Back Up**” the utility values of the leaf nodes along their corresponding branches, **combining it with the rewards** along that path

- **Pick the branch with the highest expected value**
Look Ahead Search for Optimal Policy

- **Decision** $A_t$ in $P(X_t|E_{1:t}A_{1:t-1})$

- **Observation** $E_{t+1}$

- $A_{t+1}$ in $P(X_{t+1}|E_{1:t+1}A_{1:t})$

- $|E_{t+2}$

- $A_{t+2}$ in $P(X_{t+2}|E_{1:t+2}A_{1:t+1})$

- $|E_{t+3}$

- $P(X_{t+3}|E_{1:t+3}A_{1:t+2})$

- Belief states are computed via any filtering algorithm, given the sequence of actions and observations up to that point.

- These are chance nodes, describing the probability of each observation.

- To back up the utilities:
  - Take average at chance points
  - Take max at decision points
Best action at time $t$?

A. $a_1$

B. $a_2$

C. indifferent
\( X \times x_1 \times x_2 \)

\( E = e_1 e_2 \)

\( A = a_1 a_2 \)

\[ b(x_t) = 0.52 \]

\( \text{select } a_1 \)

\( \max \)

\( e_1 \cdot 0.6 \)

\( e_2 \cdot 0.4 \)

\( e_1 \cdot 0.5 \)

\( e_2 \cdot 0.5 \)

\( U(x_1, x_2) = 0.8 \)

\( U(x_1, x_2) = 1.0 \)

\( U(x_1, x_2) = 0.1 \)

\( U(x_1, x_2) = 0.9 \)

\( U(x_1, x_2) = 0.1 \)

\( U(x_1, x_2) = 1.0 \)

\( U(x_1, x_2) = 0.0 \)

\( U(x_2) = 1 \)

\( U(x_2) = 0 \)

\( U(x_2) = 0 \)

\( U(x_2) = 0 \)

\( U(x_2) = 0 \)

\( U(x_2) = 0 \)

\( U(x_2) = 0 \)

\( U(x_2) = 0 \)

\( U(x_2) = 0 \)
Look Ahead Search for Optimal Policy

What is the time complexity for exhaustive search at depth \( d \), with \(|A|\) available actions and \(|E|\) possible observations?

A. \( O(d \cdot |A| \cdot |E|) \)
B. \( O(|A|^d \cdot |E|^d) \)
C. \( O(|A|^d \cdot |E|) \)

Would Look ahead work better when the discount factor is?

A. Close to 1
B. Not too close to 1
Finding the Optimal Policy: State of the Art

- Turn a POMDP into a corresponding MDP and then apply VI: only small models

- Generalize VI to work on POMDPs
  - 10 states in 1998
  - 200,000 states in 2008-09

- Develop Approx. Methods
  - Point-Based VI and Look Ahead

- Even 50,000,000 states
  http://www.cs.uwaterloo.ca/~ppoupart/software.html
Some Applications of POMDPs……


Another “famous” Application

Learning and Using POMDP models of Patient-Caregiver Interactions During Activities of Daily Living

Goal: Help Older adults living with cognitive disabilities (such as Alzheimer's) when they:

• forget the proper sequence of tasks that need to be completed
• they lose track of the steps that they have already completed.

Source: Jesse Hoey

UofT 2007
R&R systems BIG PICTURE

Environment

Deterministic
- Arc Consistency
  - Search
  - SLS

Stochastic

Problem

Deterministic
- Search

Stochastic
- Var. Elimination
- Approx. Inference
- Temporal. Inference
- Value Iteration
- Approx. Inference

Search

Constraint Satisfaction

Query

Static

Logics

- Vars + Constraints
  - Search

Sequential

Planning

Representation

Reasoning Technique

CPSC422, Lecture 6

Slide 22
422 big picture

**Deterministic**
- Logics
  - First Order Logics
- Ontologies
  - Temporal rep.

**Stochastic**
- Belief Nets
  - Approx. : Gibbs
- Markov Chains and HMMs
  - Forward, Viterbi....
  - Approx. : Particle Filtering

**Undirected Graphical Models**
- Conditional Random Fields

**Markov Decision Processes and Partially Observable MDP**
- Value Iteration
- Approx. Inference

**Applications of AI**

**Hybrid: Det + Sto**
- Prob CFG
- Prob Relational Models
- Markov Logics

**CPSC 422, Lecture 34**

**Slide 23**
Learning Goals for today’s class

You can:

• Define a **Policy** for a POMDP

• Describe space of possible methods for computing optimal policy for a given POMDP

• Define and trace Look Ahead Search for finding an (approximate) Optimal Policy

• Compute Complexity of Look Ahead Search
TODO for next Wed

• Read textbook 11.3 (Reinforcement Learning)
  • 11.3.1 Evolutionary Algorithms
  • 11.3.2 Temporal Differences
  • 11.3.3 Q-learning

• Assignment 1 will be posted on Connect today
  • VInfo and VControl
  • MDPs (Value Iteration)
  • POMDPs
In practice, the hardness of POMDPs arises from the complexity of policy spaces and the potentially large number of states.

Nervertheless, real-world POMDPs tend to exhibit a significant amount of structure, which can often be exploited to improve the scalability of solution algorithms.

- Many POMDPs have simple policies of high quality. Hence, it is often possible to quickly find those policies by restricting the search to some class of compactly representable policies.

- When states correspond to the joint instantiation of some random variables (features), it is often possible to exploit various forms of probabilistic independence (e.g., conditional independence and context-specific independence), decomposability (e.g., additive separability) and sparsity in the POMDP dynamics to mitigate the impact of large state spaces.
Symbolic Perseus

• Symbolic Perseus - point-based value iteration algorithm that uses Algebraic Decision Diagrams (ADDs) as the underlying data structure to tackle large factored POMDPs

• Flat methods: 10 states at 1998, 200,000 states at 2008

• Factored methods: 50,000,000 states

• http://www.cs.uwaterloo.ca/~ppoupart/software.html
By applying simple rules of probability we can derive a:

**Transition model** \( P(b' \mid a, b) \)

\[
P(b' \mid a, b) = \sum_{e} P(b' \mid e, a, b) \sum_{s'} P(e \mid s') \sum_{s} P(s' \mid s, a)b(s)
\]

where \( P(b' \mid e, a, b) = 1 \) if \( b' = \text{Forward}(e, a, b) \)

\[= 0 \quad \text{otherwise}
\]

When the agent performs a given action \( a \) in belief state \( b \), and then receives observation \( e \), filtering gives a unique new probability distribution over state.

**Deterministic transition from one belief state to the next**

- We can also define a *reward function* for belief states

\[
\rho(b) = \sum_{s} b(s)R(s)
\]
Solving POMDP as MPD

- So we have defined a POMD as an MDP over the belief states
  - Why bother?

- Because it can be shown that an optimal policy $\pi^*(b)$ for this MDP is also an optimal policy for the original POMDP
  - i.e., solving a POMDP in its physical space is equivalent to solving the corresponding MDP in the belief state

- Great, we are done!
POMDP as MDP

- But how does one find the optimal policy $\pi^*(b)$?
  - One way is to restate the POMDP as an MPD in belief state space

- **State space**:
  - space of probability distributions over original states
  - For our grid world the belief state space is?
  - initial distribution $<1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 1/9, 0, 0>$ is a point in this space

- What does the transition model need to specify?

\[ P(b' | a, b) \]
Does not work in practice

- Although a transition model can be effectively computed from the POMDP specification
- Finding (approximate) policies for continuous, multidimensional MDPs is PSPACE-hard
  - Problems with a few dozen states are often unfeasible
- Alternative approaches…. 
How to Find an Optimal Policy?

- Turn a POMDP into a corresponding MDP and then solve the MDP (😊)
- Generalize VI to work on POMDPs (also 😞)
- Develop Approx. Methods (😊)
- **Point-Based Value Iteration**
- Look Ahead
Recent Method: Point-based Value Iteration

- Find a solution **for a sub-set of all states**
- Not all states are necessarily reachable
- Generalize the solution to all states
- Methods include: PERSEUS, PBVI, and HSVI and other similar approaches (FSVI, PEGASUS)
How to Find an Optimal Policy?

- Turn a POMDP into a corresponding MDP and then solve the MDP
- Generalize VI to work on POMDPs (also 😞)
- Develop Approx. Methods (😊)
  - Point-Based VI
- Look Ahead