Dialogue as a Decision Making Process

Nicholas Roy

Challenges of Autonomy in the Real World

- Wide range of sensors
  - Noisy sensors
  - World dynamics
  - Adaptability
  - Incomplete information

- Robustness under uncertainty

Minerva

Pearl

Predicted Health Care Needs

- By 2008, need 450,000 additional nurses:
  - Monitoring and walking assistance
    - 30% of adults 65 years and older have fallen this year
  - Cost of preventable falls: $32 Billion US/year

- Intelligent reminding
  - Cost of medication non-compliance: $1 Billion US/year

Spoken Dialogue Management

- We want...
  - Natural dialogue...
    - With untrained (and untrainable) users...
    - In an uncontrolled environment...
    - Across many unrelated domains
  - Cost of errors...
    - Medication is not taken, or taken incorrectly
    - Robot behaves inappropriately
    - User becomes frustrated, robot is ignored, and becomes useless
- How to generate such a policy?
Perception and Control

Probabilistic Methods for Dialogue Management

Markov Decision Processes

- A Markov Decision Process is given formally by the following:
  - A set of states \( S = \{s_1, s_2, ..., s_n\} \)
  - A set of actions \( A = \{a_1, a_2, ..., a_m\} \)
  - A set of transition probabilities \( T(s_i, a, s_j) = p(s_j|a, s_i) \)
  - A set of rewards \( R: S \times A \rightarrow \mathbb{R} \)
  - A discount factor \( \gamma \in [0, 1] \)
  - An initial state \( s_0 \in S \)

- Bellman's equation (Bellman, 1957) computes the expected reward for each state recursively,
  \[
  J(s_i) = \max_a \left( R(s_i, a) + \gamma \sum_{j=1}^{n} T(s_i, a, s_j) \cdot J(s_j) \right)
  \]
- And determines the policy that maximises the expected, discounted reward

The POMDP in Dialogue Management

- State: Represents desire of user
e.g. want_tv, want_meds
- This state is unobservable to the dialogue system
- Observations: Utterances from speech recogniser
e.g. I want to take my pills now.
- The system must infer the user's state from the possibly noisy or ambiguous observations
- Where do the emission probabilities come from?
  - At planning time, from a prior model
  - At run time, from the speech recognition engine

The MDP in Dialogue Management

- State: Represents desire of user
e.g. want_tv, want_meds
- Assume utterances from speech recogniser give state
e.g. I want to take my pills now.
- Actions are: robot motion, speech acts
- Reward: maximised for satisfying user task

Markov Decision Processes

- Model the world as different states the system can be in
e.g. current state of completion of a form
- Each action moves to some new state with probability \( p(i, j) \)
- Observation from user determines posterior state

Observation: "Go to kitchen"

State: Ask for instructions
State: Know nothing
State: Goal-in-Kitchen
State: Goal Achieved

Action: Go to kitchen

Actions: Ask for instructions
State: Goal Achieved
Markov Decision Processes

- Optimal policy maximizes expected future (discounted) reward
- Policy found using value iteration

![Diagram](image)

Markov Decision Processes

- Since we can compute a policy that maximises the expected reward...
- then if we have ...
  - a reasonable reward function
  - a reasonable transition model
- Do we get behaviour that satisfies the user?

Fully Observable State Representation

- Advantage: No state identification/tracking problems
- Disadvantage: What if the observation is noisy or false?

Perception and Control

Talk Outline

- Robots in the real world
- Partially Observable Markov Decision Processes
  - Solving large POMDPs
  - Deployed POMDPs
Control Models

- Markov Decision Processes
- Partially Observable Markov Decision Processes

POMDPs

The POMDP in Dialogue Management

The POMDP in Dialogue Management

Navigation as a POMDP

The POMDP in Dialogue Management

What’s on NBC?
He wants the TV schedule, but I’m not sure which channel.

Sorry, which channel did you want?

The POMDP in Dialogue Management

![Image of two robots, one labeled want_AB and the other want_RC, with a speech bubble saying NBC please.

He wants the schedule for NBC!

Probability mass still distributed among multiple states, but mostly centered on the true state now.

POMDP Advantages

- Models information gathering
- Computes trade-off between: Getting reward, Being uncertain

- MDP makes decisions based on uncertain foreknowledge
- POMDP makes decisions based on uncertain knowledge

A Simple POMDP

![Image of a robot holding a bar graph with states s1, s2, and s3, and a belief space p(s).

State is hidden

POMDP Policies

![Image of a 3D graph showing belief space, current belief, and optimal action.

Belief Space

Value Function over Belief Space

Current belief

Optimal action

POMDP Policies

![Image of a 3D graph showing belief space, current belief, and optimal action.

Belief Space

Value Function over Belief Space

Current belief

Optimal action

Dateline is on NBC right now.

Want ABC

Want_CBS

Want_NBC

Want_AB

Want_RC
Scaling POMDPs

This simple 20 state maze problem takes 24 hours for 7 steps of value iteration. 

Goal

Littman et al. 1997
Hauskrecht 2000
1 hour, Zhang & Zhang 2001

The Real World

Maps with 20,000 states
600 state dialogues

Structure in POMDPs

- Factored models
  - Boutilier & Poole, 1996
  - Guespin, Koller & Parr, 2001

- Information Bottleneck models
  - Poupart & Boutilier, 2002

- Hierarchical POMDPs
  - Pineau & Thrun, 2000
  - Mahadevan & Theocharous 2002

- Many others

Belief Space Structure

The controller may be globally uncertain...
but not usually.

Belief Compression

- If uncertainty has few degrees of freedom, belief space should have few dimensions

Each mode has few degrees of freedom

Control Models

- Previous models
  - Brittle
  - Intractable

- Compressed POMDPs
The Augmented MDP

- Represent beliefs using:
  \[ \tilde{b} = \left\{ \arg\max_s b(s); H(b) \right\} \]
  \[ H(b) = \sum_{i=1}^{N} p(s_i) \log_2 p(s_i) \]
- Discretise into 2-dimensional belief space MDP

Model Parameters

- Reward function

\[ R(\tilde{b}) = \sum p(s)R(s) \]

Augmented MDP

1. Discretise state-entropy space
2. Compute reward function and transition function
3. Solve belief state MDP

Nursebot Domain

- Medication scheduling
- Time and place tracking
- Appointment scheduling
- Simple outside knowledge
  e.g. weather
- Simple entertainment
  e.g. TV schedules
- *Sphinx speech recognition, Festival speech synthesis*

MDP Graph
### An Example Dialogue

<table>
<thead>
<tr>
<th>Observation</th>
<th>True State</th>
<th>Belief Category</th>
<th>Action</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>hello</td>
<td>request</td>
<td>hello</td>
<td>say hello</td>
<td>100</td>
</tr>
<tr>
<td>what is like</td>
<td>start state</td>
<td>request</td>
<td>say request</td>
<td>-100</td>
</tr>
<tr>
<td>what state is it for?</td>
<td>get state</td>
<td>request</td>
<td>say state</td>
<td>-100</td>
</tr>
<tr>
<td>you are高い</td>
<td>what is</td>
<td>request</td>
<td>say ABC</td>
<td>-100</td>
</tr>
<tr>
<td>you are what</td>
<td>you are</td>
<td>request</td>
<td>say ABC</td>
<td>-100</td>
</tr>
<tr>
<td>What is on ABC</td>
<td>get on</td>
<td>request</td>
<td>say ABC</td>
<td>-100</td>
</tr>
<tr>
<td>yes</td>
<td>state ABC</td>
<td>request</td>
<td>say ABC</td>
<td>-100</td>
</tr>
<tr>
<td>go to the room pretty good</td>
<td>send robot</td>
<td>request</td>
<td>say robot</td>
<td>1.0</td>
</tr>
<tr>
<td>the hallway also</td>
<td>send robot</td>
<td>request</td>
<td>confirm robot place</td>
<td>0.0</td>
</tr>
<tr>
<td>go in right hallway</td>
<td>send robot</td>
<td>request</td>
<td>ask robot where</td>
<td>0.0</td>
</tr>
<tr>
<td>the hallway also</td>
<td>send robot</td>
<td>request</td>
<td>go to kitchen</td>
<td>1.0</td>
</tr>
</tbody>
</table>

### Accumulation of Reward – Simulated 7 State Domain

![Graph showing accumulation of reward over number of dialogues.](image)

- **Larger Slope == Better Performance**

### Accumulation of Reward – Simulated 17 State Domain

![Graph showing accumulation of reward over number of dialogues.](image)

### POMDP Dialogue Manager Performance

![Graph showing user data and reward accumulation.](image)

- **User 1**: 10.1
- **User 2**: 16.5
- **User 3**: 41.0

### POMDP Dialogue Manager Performance

![Graph showing time to satisfy request.](image)

- **User 1**: 1.3
- **User 2**: 1.9
- **User 3**: 1.9

- **User 2**: 0.925

- **User 3**: 0.91
POMDPs for Navigation

- Conventional trajectories may not be robust to localization error

Estimated robot position
True robot position
Goal position

Talk Outline

- Robots in the real world
- Partially Observable Markov Decision Processes
- Solving large POMDPs
- Deployed POMDPs

Belief Compression

- Belief space is a low-dimensional sub-manifold

Full Belief Space

Dimensionality Reduction

- Principal Components Analysis
Principal Components Analysis

Given belief $B \in \mathbb{R}^n$, we want $\tilde{B} \in \mathbb{R}^m$, $m \ll n$.

Collection of beliefs drawn from 200 state problem.

Many real world POMDP distributions are characterized by large regions of low probability.

PCA data likelihood:

$$- \log P(b; U\tilde{b}) = - \log N(b; U\tilde{b})$$

Data are not normally distributed.

Minimizing PCA loss function:

$$L(b, U, \tilde{b}) = \|b - U\tilde{b}\|^2$$

Equivalent to minimizing:

$$- \log P(b; \Theta) = - \log N(b; \Theta)$$

Equivalent to minimizing:

$$- \log f(b; \theta) + B_F(b \parallel g(\Theta))$$

Principal Components Analysis

- PCA data likelihood:
  \[ -\log P(b; \mathbf{U}\mathbf{b}) = -\log \text{Poisson}(b; \mathbf{U}\mathbf{b}) \]

- Use a Poisson likelihood model

  [Collins, Dasgupta & Schapire, 2000]

Different Error Functions

- Gaussian:
  \[ p(x) \propto e^{-\frac{(x-\mu)^2}{\sigma^2}} \]

- Poisson:
  \[ p(x) \propto e^{-\lambda x} \]

Solving for Bases and Parameters

- Bregman Divergence for Poisson error model:
  \[ B_F (b \parallel \mathbf{U}\mathbf{b}) = e^{\langle \mathbf{U}\mathbf{b} \rangle} - b \circ \mathbf{U}\mathbf{b} \]

Solving for Bases and Parameters

- Bregman Divergence for Poisson error model:
  \[ B_F (b \parallel \mathbf{U}\mathbf{b}) = e^{\langle \mathbf{U}\mathbf{b} \rangle} - b \circ \mathbf{U}\mathbf{b} \]
  \[ \frac{\partial B_F (b \parallel \mathbf{U}\mathbf{b})}{\partial \mathbf{U}} = \frac{\partial}{\partial \mathbf{U}} \left( e^{\langle \mathbf{U}\mathbf{b} \rangle} - b \circ \mathbf{U}\mathbf{b} \right) \]
  \[ = e^{\langle \mathbf{U}\mathbf{b} \rangle} \mathbf{b} - \mathbf{b} \mathbf{b}^T \]
  \[ \frac{\partial B_F (b \parallel \mathbf{U}\mathbf{b})}{\partial b} = \frac{\partial}{\partial b} \left( e^{\langle \mathbf{U}\mathbf{b} \rangle} - b \circ \mathbf{U}\mathbf{b} \right) \]
  \[ = \mathbf{U}^T e^{\langle \mathbf{U}\mathbf{b} \rangle} - \mathbf{U}^T b \]

Solving for Bases and Parameters

- Loss function for Poisson error model:
  \[ -\log(x; e^{\lambda}) \propto e^x - x\lambda \]
  \[ \arg\min -\log(b; \mathbf{U}\mathbf{b}) = \arg\min e^{\langle \mathbf{U}\mathbf{b} \rangle} - b \circ \mathbf{U}\mathbf{b} \]
  \[ \text{Equivalent to minimising:} \]
  \[ \arg\min \| D^{-1/2} (b - \exp(\mathbf{U}\mathbf{b})) \| \]

Example EPCA

- Probability of being in state

11
Example Reduction

E-PCA will indicate appropriate number of bases, depending on beliefs encountered.

Finding Dimensionality

Discrete belief space MDP

Planning

Original POMDP

Low-dimensional belief space

Model Parameters

Reward function

\[ R(\tilde{b}) = E_b (R(s)) = \sum p(s)R(s) \]

Model Parameters

Transition function

\[ T(\tilde{b}_i, a, \tilde{b}_j) = ? \]
Model Parameters

\[ \text{Use forward model} \]

\[
\begin{align*}
    \mathbf{b}_i & \rightarrow \mathbf{z}_i \\
    \mathbf{b}_j & \rightarrow \mathbf{z}_j \\
    \mathbf{a}, \mathbf{z}_0 & \rightarrow \mathbf{p}(s) \\
    \mathbf{p}(s) & \rightarrow \mathbf{b}_i
\end{align*}
\]

Low dimension

Full dimension

\[ \text{Deterministic process} \]

\[ \text{Stochastic process} \]

\[ T(\mathbf{b}_i, \mathbf{a}, \mathbf{b}_j) \propto p(\mathbf{z}|s)\mathbf{b}_i(s|a) \]

if \( \mathbf{b}_j(s) = \mathbf{b}_i(s|a, z) \)

\[ = 0 \]

otherwise

E-PCA POMDPs

1. Collect sample beliefs
2. Find low-dimensional belief representation
3. Discretize
4. Compute reward function and transition function
5. Solve belief state MDP

Robot Navigation Example

Initial Distribution

True robot position

Goal state

True robot position

Goal position

People Finding as a POMDP

Factored state space

- 2 dimensions: fully-observable robot position
- 6 dimensions: distribution over person positions

Regular grid gives \(~10^{10}\) states
Variable Resolution Discretization

- Variable Resolution Dynamic Programming (1991)
- Parti-game (Moore, 1993)
- Variable Resolution Discretization (Munos & Moore, 2000)
- POMDP Grid-based Approximations (Hauskrecht, 2001)
- Improved POMDP Grid-based Approximations (Zhou & Hansen, 2001)

Variable Resolution

- Non-regular grid using samples

\[ T(b_1, a_1, b_2) \]

- Computer model parameters using nearest-neighbour

Refining the Grid

- Sample beliefs according to policy
- Construct new model
- Keep new belief if \( V(b'_1) > V(b_1) \)

The Optimal Policy

Original distribution

Reconstruction using EPCA and 6 bases

Robot position

True person position

Policy Comparison

Average time to find person

- True MDP
- Closest
- Densest
- Maximum Likelihood
- E-PCA
- Var. Res. E-PCA

E-PCA: 72 states
Var. Res. E-PCA: 260 states
## Summary

- POMDPs for robotic control improve system performance
- POMDPs can scale to real problems
- Belief spaces are structured
  - Compress to low-dimensional statistics
  - Find controller for low-dimensional space

## Open Problems

- Better integration and modelling of people
- Better spatial and temporal models
- Integrating learning into control models
- Integrating control into learning models