Dialogue Management using Partially Observable Markov Decision Processes

Jason D. Williams

Machine Intelligence Laboratory
Cambridge University Engineering Department
Outline

• Model: Proposed POMDP representation of dialogue
• Part 1: Comparison with baselines
  – Example: 3-place problem
• Part 2: Scaling up with the “summary POMDP” method
  – Example: 1000-place problem
• Conclusions & future work

Material in this talk is joint work with Pascal Poupart and Steve Young


Idealized Human-Computer Dialog

- Current state = dialogue modelling
- System action selection = dialogue management

- Dialogue state is unobserved:
  - User’s goal
  - User’s (real) action
  - Conversation state

- Inferences via observation:
  - May contain errors
Decompose state variable into 3 “models”

User beliefs

Time t

System state

User action

System action

Belief state

Observation

Time t+1

User beliefs

System state
What’s really different?

Typical approaches encode uncertainty through explicit state variables. Eg.

<table>
<thead>
<tr>
<th>Num Pizza’s:</th>
<th>Variable Status:</th>
<th>Confidence:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>undefined</td>
<td></td>
</tr>
</tbody>
</table>

Sys: How many pizza’s?
User: Three, please.

\[ A_s = \Pi(S) \]

<table>
<thead>
<tr>
<th>Num Pizza’s:</th>
<th>Variable Status:</th>
<th>Confidence:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>defined</td>
<td>medium</td>
</tr>
</tbody>
</table>

Sys: You want two pizza’s?
User: No.

<table>
<thead>
<tr>
<th>Num Pizza’s:</th>
<th>Variable Status:</th>
<th>Confidence:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>undefined</td>
<td></td>
</tr>
</tbody>
</table>

Sys: How many pizza’s?
User: Three, please.

\[ A_s = \Pi(S) \]

<table>
<thead>
<tr>
<th>Num Pizza’s:</th>
<th>Variable Status:</th>
<th>Confidence:</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>defined</td>
<td>medium</td>
</tr>
</tbody>
</table>

Sys: You want three pizza’s?
User: Yes.

Instead we can maintain a distribution over states – i.e., maintain all possible states – and update the distribution at each timestep

\[ A_s = \Pi(P(S)) \]

We can think of a belief state as a “cumulative” confidence measure over all user goals.
Toward dialogue management: control

Problem: how to choose $a_s$?
- Treat as a design problem (for human designers)
- Specify $r$; choose $a_s$ to max. immediate reward (no planning)
- Specify $r$; choose $a_s$ to max. cumulative reward (planning - POMDP)
A policy as a partitioning

- A policy is a mapping: situation \(\rightarrow\) action
- One representation is a \textit{partitioning} of belief space
- 2-dimensional example:
  - Simple state space with 2 user goals: A & B
  - Belief space can be written as \(b(A)\) (b/c \(b(B) = 1 - b(A)\))
  - Policy shows action to take for each point in belief space

\[
A_s = \Pi(b(S))
\]

This partitioning is produced by a POMDP optimization method
Outline

• Motivation
• Model: Proposed POMDP representation of dialogue
• Part 1: Comparison with baselines
  – Example: 3-place problem
• Part 2: Scaling up with the “summary POMDP” method
  – Example: 1000-place problem
• Conclusions & future work
3-place problem

User is travelling from $x$ to $y$ in a world with 3 cities, $a$, $b$, and $c$.

- $s_u$: The user’s desired from and to cities – e.g., $(a, b)$
- $a_u$, o: yes, no, null, a, from-a, to-a, from-a-to-b, etc.
- $s_d$: Each slot can be not stated, unconfirmed, or confirmed
- $a_s$: greet, ask-from, ask-to, conf-from-a, conf-to-b, submit-a-b, fail

- **User goal model**: User has a fixed goal throughout the dialogue
- **User action model**: User responds with varied but cooperative actions. Sometimes the user doesn’t respond ($null$)
- **Conversation model**: Deterministically “promotes” slots
- **Speech recognition model**: Uniform errors with probability $p_{err}$
- **Reward fn**: +10 correct submit, -10 wrong submit, -5 fail, -3 for confirming something which is not stated, -1 per-turn otherwise

- **Solve with Perseus** (PBVI)
Baseline 1: MDP

- Same system dynamics as in POMDP
- Pass observations to an MDP
- *From* and *to* slots can be “unknown”, “observed”, or “confirmed”
- State estimator using typical heuristics
  - Set a slot to “unknown” when user answers “no” to an explicit confirmation

Optimize using a standard technique (*Q-learning*)
Results: POMDP vs. MDP

Expected or average return vs. $p_{err}$

- Red line: POMDP
- Blue line: MDP
Baseline 2: handcrafted policies

- Another way to represent a policy: directed graph (FSA)
- We can evaluate the FSA & POMDP to find the expected value of executing the FSA

Example (1 of 3):
Results: POMDP vs. 3 Handcrafted policies

-8 -6 -4 -2 0 2 4 6 8 10

Expected return

0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65

p_{err}

POMDP
HC1
HC2
HC3
Outline

- Motivation
- Model: Proposed POMDP representation of dialogue
- Part 1: Comparison with baselines
  - Example: 3-place problem
- Part 2: Scaling up with the “summary POMDP” method
  - Example: 1000-place problem
- Conclusions & future work
POMDPs scale poorly

Belief space, actions, and observations cover $N$ slots and $M$ slot values

<table>
<thead>
<tr>
<th>$i$</th>
<th>$S_u$</th>
<th>$A_u$</th>
<th>$A_s$</th>
<th>$O$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>london</td>
<td>london, to(london)</td>
<td>conf(london), submit(london)</td>
<td>london, to(london)</td>
</tr>
<tr>
<td>2</td>
<td>leeds</td>
<td>leeds, to(leeds)</td>
<td>conf(leeds), submit(leeds)</td>
<td>leeds, to(leeds)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>..., to(...)</td>
<td>conf(...), submit(...)</td>
<td>... , to(...)</td>
</tr>
<tr>
<td>$M$</td>
<td>dover</td>
<td>dover, to(dover)</td>
<td>conf(dover), submit(dover)</td>
<td>dover, to(dover)</td>
</tr>
</tbody>
</table>

For $M=1000$ and $N=2$, there are $\sim 10^6$ states, actions, and observations

Growth factor is $O(M^N)$

Can perform belief monitoring with factoring, but how to optimize?
Intuition of the “Summary POMDP”

\[ b(s_u) \]
\[ b(a_u) \]
\[ b(s_d) \]
\[ b(\tilde{s}_u) \]
\[ b(\tilde{s}_d) \]

\[ |S_u| \cdot |A_u| \cdot |S_d| = 1000 \cdot 2000 \cdot 3 \]

\[ |\tilde{S}_u| \cdot |\tilde{S}_d| = 2 \cdot 3 \]
Actions in the summary POMDP

\[
\begin{align*}
\alpha & \quad \tilde{\alpha} \\
\text{ask} & \quad \text{ask} \\
\text{confirm(london)} & \quad \text{confirm(x)} \\
\text{confirm(leeds)} & \quad x = \arg \max (b(s_u)) \\
\text{confirm(dover)} & \quad \text{submit(x)} \\
\ldots & \quad x = \arg \max (b(s_u)) \\
\text{submit(london)} & \quad \text{submit(x)} \\
\text{submit(leeds)} & \quad x = \arg \max (b(s_u)) \\
\text{submit(dover)} & \quad \text{submit(x)} \\
\ldots & \quad \text{submit(x)}
\end{align*}
\]
Observations in the summary POMDP

Observations in summary POMDP give an compact indication of how \( b(\tilde{s}) \) is changing based on \( s \) and \( o \). It has 2 parts.

\[
\arg \max (b(s_u)) = \arg \max (b(s'_u))?
\]

\( \tilde{o}_1 \in \{\text{same, different}\} \)

Is \( \arg \max (b(s_u)) \) "consistent" with \( o \)?

\( \tilde{o}_2 \in \{\text{consistent, inconsistent, no–info}\} \)
“Summary POMDP” method

1. Sample
   \[ p(s' | s, a) \]
   \[ p(o' | s', a) \]
   \[ r(s, a) \]

2. Optimize
   \[ p(\tilde{s}' | \tilde{s}, \tilde{a}) \]
   \[ p(\tilde{o}' | \tilde{s}', \tilde{a}) \]
   \[ r(\tilde{s}, \tilde{a}) \]
   \[ \pi(b(\tilde{s})) \]

3. Run
   \[ b(\tilde{s}_d) \leftarrow b(s_d) \]
   \[ b(\tilde{s}_u) \leftarrow \max(b(s_u)) \]
   \[ \tilde{a} \leftarrow \pi(b(\tilde{s})) \]
   \[ a \leftarrow (\tilde{a}, \text{arg max}(b(s_u))) \]

run maintain \( b(s) \)
1000-place problem

User is travelling from x to y in a world with 1000 cities

Same as 3-place problem, with the following extensions

• More factoring, to support efficient belief monitoring
• \( a_u \): can include more complex actions like \{yes, to(london)\}

• **Speech recognition model**: Substitutions & deletions with \( p_{err} \)
• **Reward fn**: Richer set of rewards for appropriateness
• **User action model**: Estimated from real dialog data (next)
2 subjects interact through a simulated speech recognition channel

Typical end-pointing model

Utterances are transcribed by a typist

Confusions generated using lexicon, phonetic confusion matrix, and language model

“SACTI-1” dialog data corpus

- “Tourist information” domain
- Wizard has more info than user; user given specific task
- 36 users / 12 wizards
- 144 dialogs
- 4071 turns
- 4 different WER targets
- Task completion
- Likert-scale questions

Conversational characteristics broadly similar to those observed in human/machine dialog.

User model example

\[ P(A_u \text{ -WH} | A_s, S_u = \text{to(london)}) \]

\[ A_s \]

Where are you going to?
To London, is that right?
Where are you going from?

P mass of A_u ("to/WH" portion only)

<table>
<thead>
<tr>
<th>&quot;London&quot;</th>
<th>&quot;To London&quot;</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data was divided, and two model sets were created:
Training model set & Testing model set
### Turn

<table>
<thead>
<tr>
<th><strong>Where to?</strong></th>
<th>$b(s_{\text{from}})$</th>
<th>$\tilde{b}(s_{\text{from}})$</th>
<th>$b(s_{\text{to}})$</th>
<th>$\tilde{b}(s_{\text{to}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>London</strong></td>
<td>CAMB.</td>
<td>BEST</td>
<td>LEEDS</td>
<td>BEST</td>
</tr>
<tr>
<td></td>
<td>SHEFF.</td>
<td>REST</td>
<td>OXFORD</td>
<td>REST</td>
</tr>
<tr>
<td></td>
<td>LONDON</td>
<td></td>
<td>LONDON</td>
<td></td>
</tr>
<tr>
<td><strong>[Leeds]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Where from?</strong></td>
<td>From Cambridge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>From Cambridge</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>[To Oxford]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Where from?</strong></td>
<td>From Cambridge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>From Cambridge</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>To Leeds, right?</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>No, to London</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>[No, from Sheffield]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Where to?</strong></td>
<td>London from Cambridge</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>London from Cambridge</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Performance vs. $p_{err}$

![Graph showing the relationship between $p_{err}$ and reward gained/turn. The graph indicates a decreasing trend in reward as $p_{err}$ increases. The user model built from the training set is represented by a blue line with error bars.](image-url)
Performance vs. $p_{err}$

- User model built from training set
- User model built from test set
Scalability vs. direct optimization

Average or expected return

$M$ (Number of distinct slot values)

Summary POMDP
Baseline
Outline

• Motivation
• Model: Proposed POMDP representation of dialogue
• Part 1: Comparison with baselines
  – Example: 3-place problem
• Part 2: Scaling up with the “summary POMDP” method
  – Example: 1000-place problem
• Conclusions & future work
Conclusions & future work

• For slot-based dialog management, POMDPs…
  – Outperform MDP baseline
  – Outperform 3 handcrafted baselines
  – Can be scaled as a “summary POMDP” to many slot values

• Interesting future avenues
  – Scale *number of* slots to 10s
  – Move beyond slot-based approaches
  – Apply to real system
Thanks!

Jason D. Williams
jdw30@cam.ac.uk

Joint work with
Pascal Poupart
ppoupart@cs.waterloo.ca

Steve Young
sjy@eng.cam.ac.uk