Reinforcement learning for spoken dialog systems: Using POMDPs for Dialog Management

Steve Young

Cambridge University Engineering Department
Machine Intelligence Laboratory
Outline of Talk

- the promise of statistical dialog systems
- Markov Decision Processes and their limitations
- Partially Observable MDPs – an intractable solution?
- the Hidden Information State system – a proof of concept.
A statistical approach to dialog system design offers the following potential advantages:

- formalise dialog design criteria as objective reward functions
- automatically learn dialog strategies from data
- allow decision making to be optimised
- increase robustness to recognition/understanding errors
- enable on-line dialog policy adaptation to allow the system to learn from experience

Overall, increase robustness and reduce design, implementation and maintenance costs

Markov Decision Processes provide the framework to do this .....
Dialog as a Markov Decision Process


Training an MDP

Key idea is to associate a value function with each state

\[ V^\pi(s_m) = E_\pi \{R \mid s_m\} \quad Q^\pi(s_m, a_m) = E_\pi \{R \mid s_m, a_m\} \]

where \( Q^\pi(s_m, \pi(s_m)) = V^\pi(s_m) \)

Given \( V \) or \( Q \), policy optimisation is straightforward since if

\[ Q^\pi(s_m, \pi'(s_m)) > V^\pi(s_m) \]

then policy \( \pi' \) is better than policy \( \pi \).

A popular algorithm for implementing this is Q-Learning
Limitations of MDP Framework

Modelling dialog as an MDP suffers from a variety of practical problems:

- State space is huge, hence propositional content and much of the relevant history is often ignored.
- Dialogs are fragile because user state $s_u$ and user dialog act $a_u$ are uncertain, hence estimate of machine state $\tilde{s}_m$ is often incorrect.
- Recovery strategies are difficult since no information is available for backtracking.
- No principled way to handle N-best ASR output.
Dialog as a Partially Observable MDP

Distribution over possible dialog acts (eg N-best list)

Distribution over all possible machine states

Policy now depends on state distribution not just the most likely state


Belief Update Equation

Belief is updated every dialog turn as follows:

\[
b'(s'_m) = k.P(o' | a'_u)P(a'_u | s'_m, a_m) \sum_{s_m} P(s'_m | s_m, a_m).b(s_m)
\]
Robustness of POMDP vs. MDP

Simulation of simple 2 slot 3-city travel problem

Expected or average return

$\rho_{err}$

Summary of the POMDP Framework

- System maintains multiple dialog hypotheses called the belief state
- Machine actions are based on the full belief state distribution not just the most likely state
- No backtracking is required when misunderstanding detected
- Speech understanding output is regarded as an observation
- Belief distribution is re-computed each time a new observation is received in a process called belief monitoring
- N-best ASRU outputs naturally incorporated into belief monitoring framework via an observation model
- POMDP framework naturally includes a user model which gives probability of each user act given each possible dialog hypothesis

However, there are some issues ....
Belief Monitoring

But representation of these distributions in a practical system is unclear.
Consider a system with just two states and 3 actions

POMDP value functions are hyperplanes in belief space. Upper surface provides defines the value function $V(b)$. Exact learning is iterative and **effectively intractable.**
Scaling to Real Systems

- POMDPs provide an elegant mathematical framework for modelling spoken dialog systems but ....
- State space will be huge – direct belief monitoring is impractical.
- Exact POMDP optimisation is intractable - even approximate POMDP optimisation is limited to a few thousand states

A solution – the *Hidden Information State* Dialog Model
The Hidden Information State Model

The HIS model provides a scaleable POMDP framework for implementing practical spoken dialog systems.

- Partition state space and compute partition beliefs *not* state beliefs
- Represent user goals by branching-tree driven by ontology rules.
- Maintain two state spaces: master space and summary space. Monitor beliefs in master space, apply and optimise policies in summary space.
- Use grid-based approximations, hence finite policy table.
Structure of a HIS Dialog Hypothesis

\[ h = \langle \{s_u\}, a_u, s_d \rangle \]

- **User goal** – tree structured set of entities
- **User action** – a dialog type plus goal tree bindings
- **Dialog history** – grounding status of each tree node

A single hypothesised information state

User goal tree built incrementally from rules, expanded on demand to accommodate user dialog acts
HIS Partitions

- Each partition represents a group of user goal states.
- Partitions are stored as tree structures, with nodes defined by a task ontology.
- Partitions are split by incoming user dialog acts.
- When a partition is split, its belief is shared between the splits.

Example ontology rules:

- entity -> venue(name, type, area) 1.0
- type -> bar(drinks, music) 0.4
- type -> restaurant(food, price) 0.3
- area -> (central | east | west | ....)
- food -> (Italian | Chinese | ....)

Structure rules with prior probs
Lexical/Dbase rules
Incoming dialog acts cause partitions to be extended and split in order to match the items in the dialog act with the nodes in the tree.

- **entity** -> venue(name,type,area)
- **type** -> bar(drinks,music) 0.4
Master <-> Summary State Mapping

Master space is mapped into a reduced summary space:

\[
\begin{align*}
\text{find(venue(hotel,area=east,near=Museum))} \\
\text{find(venue(bar,area=east,near=Museum))} \\
\text{find(venue(hotel,area=east))} \\
\text{find(venue(hotel,area=west))} \\
\text{find(venue(hotel))} \\
\text{....etc}
\end{align*}
\]
The POMDP Policy and Action Selection

A set of points in summary space and their associated actions

\[ \hat{a}_m, \hat{b}_t, \hat{a}_{m,t} \]

Policy \( \pi \)

Find nearest belief point

Action selection at time \( t \)
Policy Optimisation

- Use Q-learning with a simulated user on belief points
- Start with a single belief point
- Add new points as they are encountered up to some maximum

![Graph showing average reward over training dialogs](x1000 Training Dialogs)
Summary of HIS Dialog Manager Operation

1. Grow Forest i.e. extend partitioning of Belief Space

2. Bind User Acts to Partitions

3. Update Dialog History

4. Form New Hypotheses

5. Map from master space -> summary space

6. Apply policy in summary space

7. Map back to master space

Heuristic Action

Refinement

Strategic Action

POMDP Policy

Map

Application Database

Belief State

Observation

From User

Ontology Rules

From System

Summary Space

Belief State

Summary of HIS Dialog Manager Operation

© Steve Young, 2006
Evaluation

The system was tested by human users in a two day study conducted simultaneously at Edinburgh and Cambridge.

Dialogues were deemed to be successfully completed when the system made a correct recommendation.

<table>
<thead>
<tr>
<th></th>
<th>Cambridge</th>
<th>Edinburgh</th>
<th>Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td># subjects</td>
<td>23</td>
<td>17</td>
<td>40</td>
</tr>
<tr>
<td># dialogues</td>
<td>92</td>
<td>68</td>
<td>160</td>
</tr>
<tr>
<td>% WER</td>
<td>21.1</td>
<td>37.3</td>
<td>29.3</td>
</tr>
<tr>
<td>% completion rate</td>
<td>95.7</td>
<td>83.8</td>
<td>90.6</td>
</tr>
<tr>
<td>Average turns to</td>
<td>3.8</td>
<td>8.1</td>
<td>5.6</td>
</tr>
<tr>
<td>completion</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Work supported by the EU FP6 “TALK” Project
Partially observable MDPs provide a natural framework for modelling spoken dialog systems:

- explicit representation of uncertainty
- support for N-best ASR output
- incorporates user and observation model
- simple error recovery by shifting belief to alternative hypotheses
- potential for on-line adaptation

The Hidden Information State system demonstrates that POMDPs can be scaled to handle real world tasks

There are many issues to resolve e.g. effective observation and user models, choice of summary state mapping, improved training procedures ...

... but overall POMDPs provide an opportunity for making significant improvements to both the design and implementation of spoken dialog systems.