Dialog Management

Spoken Dialog Study Group

by: Lasguido Nio, D2 Student

Augmented Human Communication Laboratory; Nara Institute of Science and Technology, Japan
Outline

• Statistical approaches to dialogue control
  – Reinforcement learning in optimizing dialogue control
    • MDP
    • POMDP
  – Some issues in reinforcement learning application on dialogue management

• Advance approach to dialogue management
  – Information state approach
  – Plan-based approach
  – Case study: DIHANA System
Statistical approach to dialogue control

Introduction (1)

Dialog control:
• mainly implemented with handcrafted rules

Issue in optimal spoken dialog system:
• purpose of the system
• intended users
• usage cost

In general, optimal system is one that completes the transaction successfully and efficiently.
Statistical approach to dialogue control

Introduction (2)

• There is a trade off between *transaction success* and *transaction efficiency*.

• Handcraft dialogue control requires an expert designer.

• Another way to manage dialog control is using data-driven approach:
  – Supervised learning: (Error handling section) require corpus that provide example of optimal decision
  – Utility maximization: System choose a action with the maximum utility immediately
  – Reinforcement learning: System choose a action that maximizes the sum of utilities over time
Reinforcement learning

- Reward: utilities of a dialog
- Return: the sum of dialog utilities over time
- Idea of RL: Specify a priorities of dialogue system in a reward function.

In RL, designer specify the desired outcomes and the RL will learn the dialog strategies details.
An application of reinforcement learning: Markov Decision Process (MDP)

Dialog formalization:

• \( S \): a set of system states
• \( A \): a set of action, to change a state
• \( T \): set of transition probabilities \( P_T(S_t|S_{t-1}, a_{t-1}) \)
• \( R \): reward when taking a particular action
• \( \pi \): a policy, mapping between state space and the action set

In conventional system:
Behaving optimally means DM must select an action in each state that has the maximum reward (utility maximization)
An application of reinforcement learning: Markov Decision Process (MDP)

• In MDP, best dialog strategy is learned by computing an optimal policy \( \pi \) that maximizes the sum of rewards (return).

• RL is used to explore and select the best policy systematically, by using empirical data such as interaction of real/simulated users with the system.

• The performance function used is usually a combination of several measurement. For example: user satisfaction, number of database access, dialogue duration, ASR error, etc..
Markov Decision Process (MDP)
Study case: NJFun system

- NJFun is a dialog system that helps people choose recreational activities in New Jersey (Singh et al., 2002)

- In NJFun, a relevant dialog decision involves:
  - Choice of dialogue initiative
  - System prompt
  - Grammar type
  - Confirmation strategy

- The range of system initiative when is defined as:
  - System initiative: directive prompt, restricted grammar
  - User initiative: open prompt, nonrestrictive grammar
  - Mixed initiative: directive prompt, nonrestrictive grammar
Markov Decision Process (MDP)
Study case: NJFun system > attributes

Create a dialogue vector
[greet, attribute, confidence, value, times, grammar, history]

• greet: greeting flag (not yet, done)
• confidence: ASR confidence score
• attribute: keeps track of task-based information
• value: value of each attribute (unknown, known)
• times: difficulty measurement (how many times, DM asked about the attribute)
• grammar: grammar choice of the system (restrictive, nonrestrictive)
• history: save a history of previous dialogue (problematic, successful)
Markov Decision Process (MDP)
Study case: NJFun system > attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Values</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greet (G)</td>
<td>0, 1</td>
<td>0: not yet, 1: done</td>
</tr>
<tr>
<td>Confidence (C)</td>
<td>0, 1, 2, 3, 4</td>
<td></td>
</tr>
<tr>
<td>Attribute (A)</td>
<td>1, 2, 3, 4</td>
<td>1: Activity, 2: Location, 3: Time, 4: Date</td>
</tr>
<tr>
<td>Value (V)</td>
<td>0, 1</td>
<td>0: unknown/undefined, 1: known</td>
</tr>
<tr>
<td>Times (T)</td>
<td>0, 1, 2</td>
<td></td>
</tr>
<tr>
<td>Grammar (M)</td>
<td>0, 1</td>
<td>0: restrictive, 1: nonrestrictive</td>
</tr>
<tr>
<td>History (H)</td>
<td>0, 1</td>
<td>0: problematic, 1: successful</td>
</tr>
</tbody>
</table>
Markov Decision Process (MDP)
Study case: NJFun system > dialogue practice

S1 : Welcome to NJFun. How may I help you?

U1 : I’d like to find um winetasting in Lambertville in the morning.
(ASR: I’d like to find out wineries the in the Lambertville in the morning.)

S2 : Did you say you are interested in Lambertville?

U2 : Yes.

S3 : Did you say you want to go in the morning?

U3 : Yes.

S4 : I found a winery near Lambertville that is open in the morning.
It is the Poor Richard’s Winery in Lambertville.
Thank you for using the system.
Please give me feedback by saying ‘good’, ‘so-so’, or ‘bad’

U4 : Good.

S5 : Goodbye and have a nice day.

<table>
<thead>
<tr>
<th>State* [GC AVT MA]</th>
<th>Action</th>
<th>Turn</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 00 100 00</td>
<td>GreetU</td>
<td>S1</td>
<td>0</td>
</tr>
<tr>
<td>2 12 110 00</td>
<td>NoConf</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>3 12 210 01</td>
<td>ExpConf2</td>
<td>S2</td>
<td>0</td>
</tr>
<tr>
<td>4 12 310 01</td>
<td>ExpConf3</td>
<td>S3</td>
<td>0</td>
</tr>
<tr>
<td>5 10 400 00</td>
<td>Tell</td>
<td>S4</td>
<td>1</td>
</tr>
</tbody>
</table>

*space in state for clarity
Markov Decision Process (MDP)
Study case: NJFun system > analysis (1)

The previous example shows: the learned policy uses a optimal strategy that allow a user initiative in the beginning of dialogue, than back of to mixed or system initiative.

Comparison with other fixed strategy (best practice):
• Always do system initiative and does not confirmation
• Always do user initiative and does confirmation

Shows that the learned policy perform better.
Markov Decision Process (MDP)  
Is it enough?

- MDP always assume that the system state are fully observable
- Even worst, the system assume that the current state is always right

We can extend the model by belief state, which contain a probability distribution over dialogue states. This technique is called Partially Observable Markov Decision Processes (POMDP).

Why POMDP (Williams and Young, 2007):  
Deals with the ASR and SLU uncertainties, by maintaining the $n$-best list from ASR and SLU as multiple hypotheses. Each time a new observation is received, the belief distribution is recalculated.
Applying POMDP’s to Dialogue Systems: An Example (1)

- POMDP handle a large space of possible belief states by do a policy optimization.

- Study case (Thomson et al., 2007): POMDP reward is set based on task completion (+ reward) and the number of dialogue turn (- reward).

- Experiment shows that POMDP system outperformed the handcrafted system in terms of error handling.
Applying POMDP’s to Dialogue Systems: An Example (2)

• Another study case (Williams, 2007) involve a troubleshooting dialogue system that advice users how to fix malfunctioning equipment.
  – The system face a uncertainties problem, which user may misinterpreted the equipment (example: light colors/buttons).
  – The system cannot know the true state of the product nor of the user’s action.
<table>
<thead>
<tr>
<th>ID</th>
<th>P(allOK)</th>
<th>P(pwrOn)</th>
<th>Transcript</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1*</td>
<td>56%</td>
<td>98.0%</td>
<td>I’m going to try pinging your DSL modem from here.</td>
</tr>
<tr>
<td>S2</td>
<td>-</td>
<td>-</td>
<td>There was no response so it seems that your connection isn’t working yet.</td>
</tr>
<tr>
<td>S3*</td>
<td>14%</td>
<td>96.1%</td>
<td>Please turn the DSL modem on.</td>
</tr>
<tr>
<td>U4</td>
<td>-</td>
<td>-</td>
<td>ok [ok, 0.004]</td>
</tr>
<tr>
<td>S5*</td>
<td>30%</td>
<td>96.4%</td>
<td>What color is the network light on the DSL modem?</td>
</tr>
<tr>
<td>U6</td>
<td>-</td>
<td>-</td>
<td>dontunterstand [ok, 0.452]</td>
</tr>
<tr>
<td>S7*</td>
<td>30%</td>
<td>93.8%</td>
<td>Please turn the DSL modem on.</td>
</tr>
<tr>
<td>U8</td>
<td>-</td>
<td>-</td>
<td>dontunterstand [ok, 0.454]</td>
</tr>
<tr>
<td>S9*</td>
<td>31%</td>
<td>99.1%</td>
<td>I’m going to check if there are any outages reported, just a moment.</td>
</tr>
<tr>
<td>S10</td>
<td>-</td>
<td>-</td>
<td>There are no outages reported.</td>
</tr>
<tr>
<td>S11</td>
<td>39%</td>
<td>99.1%</td>
<td>I’m going to try resetting your connection from here, just a moment.</td>
</tr>
<tr>
<td>S12</td>
<td>-</td>
<td>-</td>
<td>Ok, that has been done.</td>
</tr>
<tr>
<td>S13</td>
<td>48%</td>
<td>99.1%</td>
<td>What color is the network light on the DSL modem?</td>
</tr>
<tr>
<td>U14</td>
<td>-</td>
<td>-</td>
<td>red [red, 0.678]</td>
</tr>
<tr>
<td>S15</td>
<td>3%</td>
<td>99.8%</td>
<td>Please open the DSL configuration screen.</td>
</tr>
</tbody>
</table>

*: the transition is not available in the baseline handcraft system

P(allOK): probability that DSL connectivity is working

P(pwrOn): probability that the power to the DSL modem is on
Some issues in RL application on dialogue management

1. Designing the state space that sufficiently rich to support the learning of an accurate model, while minimizing data sparseness.
2. The amount of data required for training and testing.
3. Comparing RL-based dialogue management strategies with those using handcrafted rules is difficult.
4. Dilemma in defining a reward.
5. Tractability of reinforcement learning in MDP/POMDP
So far ...

- Statistical approaches to dialogue control
  - Reinforcement learning in optimizing dialogue control
    - MDP
    - POMDP
  - Some issues in reinforcement learning application on dialogue management

- Advance approach to dialogue management
  - Information state approach
  - Plan-based approach
  - Case study: DIHANA System
Information state approach to dialogue management: Introduction

• One way to model a dialogues is by specifying them as information states
• Information states is updated according in the course of dialogue
• This approach is similar to frame-based dialogue control model, except it includes the mental states of the speakers
Information state approach to dialogue management: Implementation

• The TrindiKit (Larsson and Traum, 2000)
  – Prolog based toolkit
  – GODIS
  – Pure implementation of Question Under Discussion (QUD)

• DIPPER (Bos et al., 2003)
  – Specific language which support declarative statement
  – Implements OAA (Open Agent Architecture)
Information state approach to dialogue management: In details (1)

- By updating the dialogue state, we can achieve dialogue dynamics

- Dialogue moves: trigger for updates, which implemented using *update rule* and *update strategy*

- Update rule: specify a rule how the information state is updated when a *dialog move* is performed. A collection of update rules is maintained by dialogue manager.
Information state approach to dialogue management: In details (2)

- Mental states of the speaker: beliefs and intentions.
- Dialogue context: dialogue act, grounded concept, and salient discourse referents.
- An information state in the GODIS

This information states can be view as a recursive features.
Information state approach to dialogue management: Summary

• This approach focus on the notion of common ground, in building a shared understanding.

• Powerful approach for modeling complex issues in dialogues

• Support complex applications and flexible presentation of the task structure, because it make a distinction between discourse context, common ground, beliefs, and intentions.
Plan-based approach to dialogue management: Introduction

Motivation: Agents are attempting to achieve a goal.

The main driving force for dialogues is planning and speaker’s plan recognition.

Goal:
- Physical state: arriving at some destination
- Mental state: knowing some information

Plan-based approach is same with information state approach, except an ability to plan and recognize planning.
Plan-based approach to dialogue management: Understand by example (1)

A dialogue conversation in TRIPS system.

U1 : We need to get the woman in Penfield to Strong.
S1 : OK.
U2 : What vehicles are available?
S2 : There are ambulances in Pittsford and Webster.
U3 : OK. Use on from Pittsford.
S3 : Do you know that Route 96 is blocked due to construction?
U4 : Oh, Let’s use the interstate instead.
S4 : OK. I’ll dispatch the crew.

This dialogue solve a task to move a patient from Penfield city to Strong hospital.

Pittsford and Webster is a town name.

The system must be able to reason the blocking (S3) to the user plan.

In addition, system could also understand the user intention by the context. Example in U2, system interpreted “vehicles” as “ambulances”.
Plan-based approach to dialogue management: Understand by example (2)

- Generation Manager handles SLU and response generation.
- Behavioral Agent: plans the system's behavior.
- Task manager: manages task/domain specific knowledge and specific discourse context.
- Interpretation Manager: provides information about the state of the dialogue.

TRIPS architecture
Case study: corpus-based dialogue management (DIHANA project)

• Dialogue system of telephone access to Spanish train timetable information

• Collect data by using Wizard of Oz technique:
  – Safe state (user input > threshold)
    • Implicit information, query to database, and answer to user
    • Inquiry to the user
    • Mixed confirmation
  – Uncertain state (Dialogue Register data < threshold)
    • Explicit confirmation
    • Mixed confirmation
Case study: corpus-based dialogue management (DIHANA project) – Selecting the next system act

\[ \hat{A}_i = \arg \max_{A_i \in A} P(A_i \mid S_1, \ldots, S_{i-1}) \]

\[ \hat{A}_i = \arg\max_{A_i \in A} P(A_i \mid DR_1, \ldots, DR_{i-1}) \]