Dialogue Systems

Regina Barzilay

April 5, 2004

Types of Dialogue Systems

- System Initiative
- Mixed Initiative

A: And what time would you like to leave Boston?
C: Uh hmm I don't think there's many options for non-stop
A: Right. There's three non-stops today
C: What are they?
A: The first one ..

Designing Dialogue Systems

User-Centered Design: Study the use and task, build simulations and prototypes, and iteratively test them on the user and fix the problems

- User Studies
- Wizard-of-Oz of Experiments
- Iterative Design

Learning Dialogue Strategies: Motivation

- Dialogue systems are complex: presentation strategies, prompts, error messages, ...
- It is hard to tune all the parameters

Reinforcement Learning

- **Exploration:** Since learning is performed while interacting, the exploration of the state-action space can be dynamically controlled by the learning algorithm
- **Delayed reinforcement:** since the costs incurred in any stage of the dialog are propagated back to all preceding state-action pairs, reinforcement learning can deal with delayed feedback
- **Adaptation:** Since the system learns from interaction, it can keep adapting for slowly changing external conditions

Action Set

All possible actions a dialogue system can perform

- Question asking for the value of the day
- Question asking for the value of the month
- Composite question asking for the value of the data (day and night)
- Final action, closing the dialog and submitting the form

Act Granularity: ask a question and activate speech recognition utility

Dialogue Design as an Optimization

- Dialogue is an interactive activity
- Dialogue is always associated with cost function (goal achievement and efficiency)
  - Cost: $C = \sum W_i(C_i)$, C_i — costs for different dialog dimensions, W_i — weights
  - Dimensions: dialog duration, cost of internal processing, cost of assessing databases, user satisfaction

Goal: Select a sequence of actions that optimizes cost function

Dialogue Representation for MDP

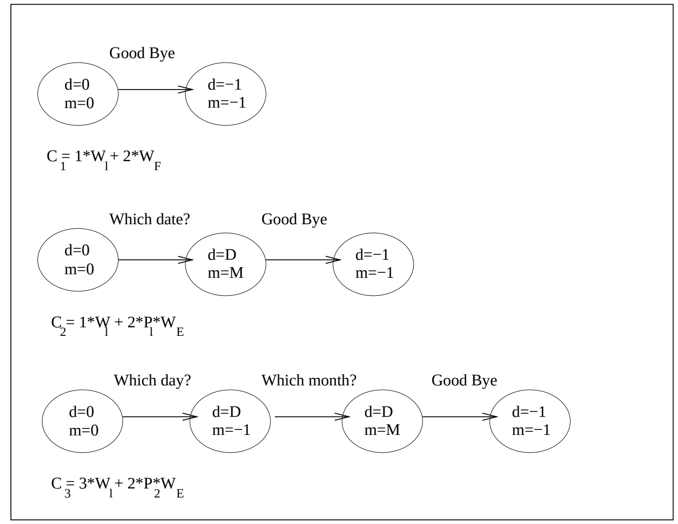
Dialog Abstraction for Markov Decision Processing

- Action Set
- State Space
- Strategy

Dialogue Strategy

Dialogue strategy specifies, for each state, reached, what is the next action invoked

Examples of Dialogue Strategies



Dialogue States

A state of a dialogue system includes the values of all the relevant information that determines the values of the next action the system executes

Example: the state description includes two integers that represent the day and the month

Total: 1+12+31+366+1

Description of Decision Precision

Initialization: $s_{t=0} = s_I$

for each interaction: T: {

if ($s_t <> s_F$) {

compute current action a_t according to the strategy

execute a_t

update current state

$t = t + 1$

}

Dialogue as MDP

$$P(s_{t+1}|s_t, s_{t-1}, \ldots, s_0, a_t, a_{t-1}, \ldots, a0) = P_T(s_{t+1}|s_t, a_t)$$

$$P(c_t|s_t, s_{t-1}, \ldots, s_0, a_t, a_{t-1}, \ldots, a0) = P_C(s_t|s_t, a_t)$$

$$C = \sum_{t=0}^{T_F} c_t$$

Optimal Strategy

- The first strategy is optimal, when recognition error is too high

$$p_2 > (W_f - W_i)/E_e$$

- The third strategy is optimal when the difference in error rates justifies a longer interactions

$$p_1 - p_2 > \frac{W_i}{2W_e}$$

Cost Distribution

p_1 — the error rate for simple questions, p_2 — the error rate for composite questions
States with one assigned variable:

$$c(s, A_f) = W_i + W_f + W_e,$$

with prob. p_2

$$c(s, A_f) = W_i + W_f,$$

with prob. $1 - p_2$

Cost Distribution

$$c(A_d) = c(A_m) = c(A_{dm}) = W_i$$

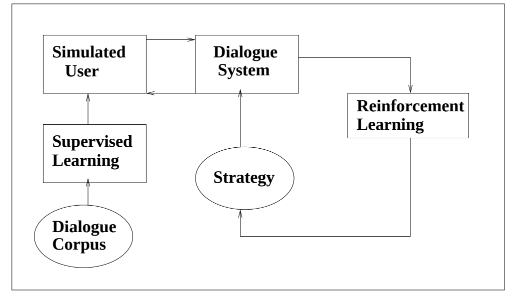
with prob.1
Final cost:

$$W_i + N_e W_e + N_f W_f,$$

N_e is a number of errors and N_f is the number of unassigned variables
Initial State: $c(S_I, A_F) = W_i + 2W_f$ with prob.1

The solution can be found using standard dynamic programming methods (value iteration, police iteration)

System Architecture



Finding the Optimal Strategy

The optimal action is selected based on the content of a given state (property of MDP)

$$V^*(s) = \sum_{t=0}^{T_F} c(s_t, a_t),$$

$c(s_t, a_t)$ is a random variable drawn from $P_T(s_{t+1}|s_t, a_t)$
The optimal value function is unique:

$$V^*(s_t) = min_a[c(s_t, a) + \sum_s P_T(s_{t+1} = s|s_t, a)V^*(s)]$$

The optimal strategy:

$$pi(s_t) = argmin_a[c(s_t, a) + \sum_s P_T(s_{t+1} = s|s_t, a)V^*(s)]$$

Simulated User

- Motivation: Expensive to train dialogue systems on real users
- Simulated User is learned from training data:
  - Rules for merging information between agent and simulated user
  - Depends only on the current action (separate modeling of act, and value distribution)

Optimal Strategy

- Start by greeting
- The system asks constraining questions
- The system retrieves data from a database
- If the result is empty, the system asks relaxation questions and then retrieves
- The system output an output, and closes the conversation

Implementation Details

- **Objective Function**:

  $$C = W_i(N_i) + W_r(N_r) + W_0(f_0(N_0)) + W_s(F_s),$$

  N_i – time of interaction, N_r – number of tuples retrieved, $f_0(N_0)$ – data presentation cost, F_s – measure of success
- **Actions**: Greeting, Constraining, Retrieval
- **State Representation**: user template, data template, system template