Efficient Model Learning for Dialog Management

Finale Doshi
Nick Roy

CSAIL

Outline

- How can Partially Observable Markov Decision Processes (POMDPs) help handle the uncertainties in dialog management?
- How can we learn user dialog models online through human-robot interactions?
  - Planning with expected parameter values
  - Planning with parameters as hidden state
  - Planning with meta-actions

Why dialog management?

•Wouldn't it be nice if we could simply tell a robot

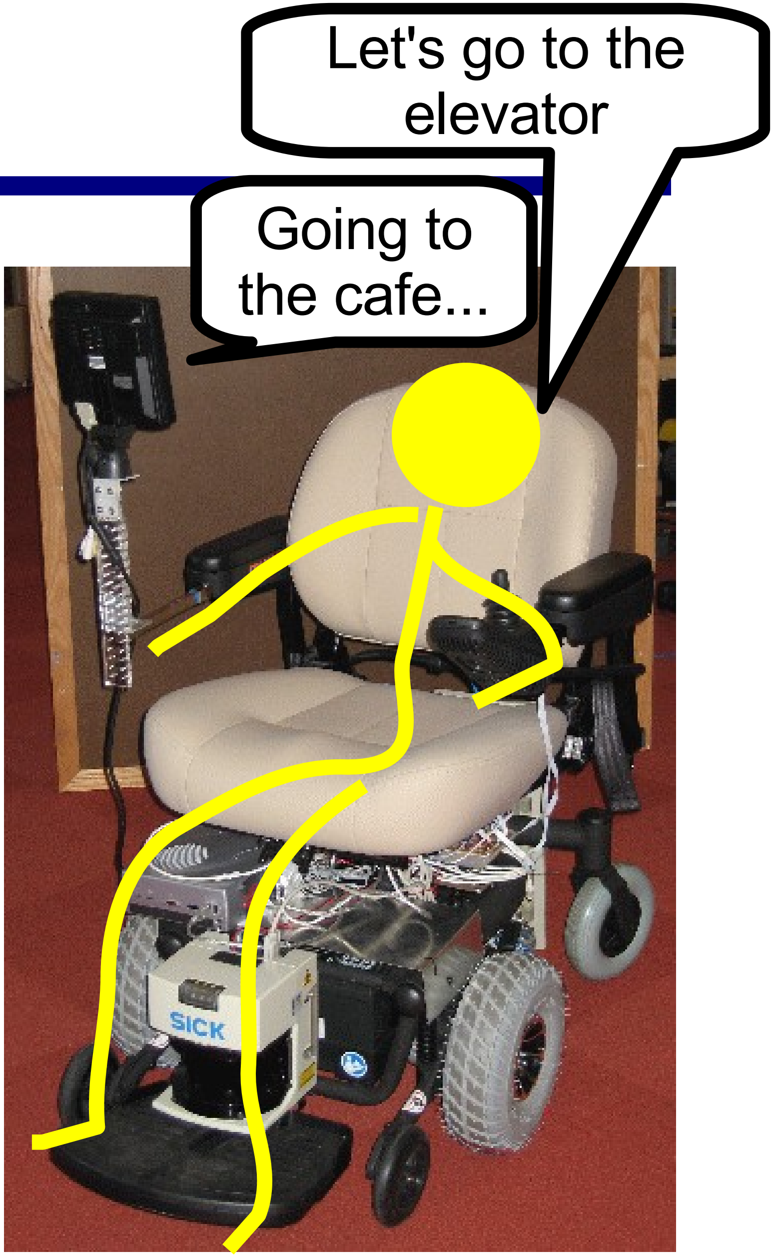
- to go to a particular location?
- to follow alongside someone?
- to remember the name of a new room?

This is would be particularly useful to wheelchair users with severely limited mobility.

The Problem

Spoken language allows for natural human robot interaction, but there are several challenges:

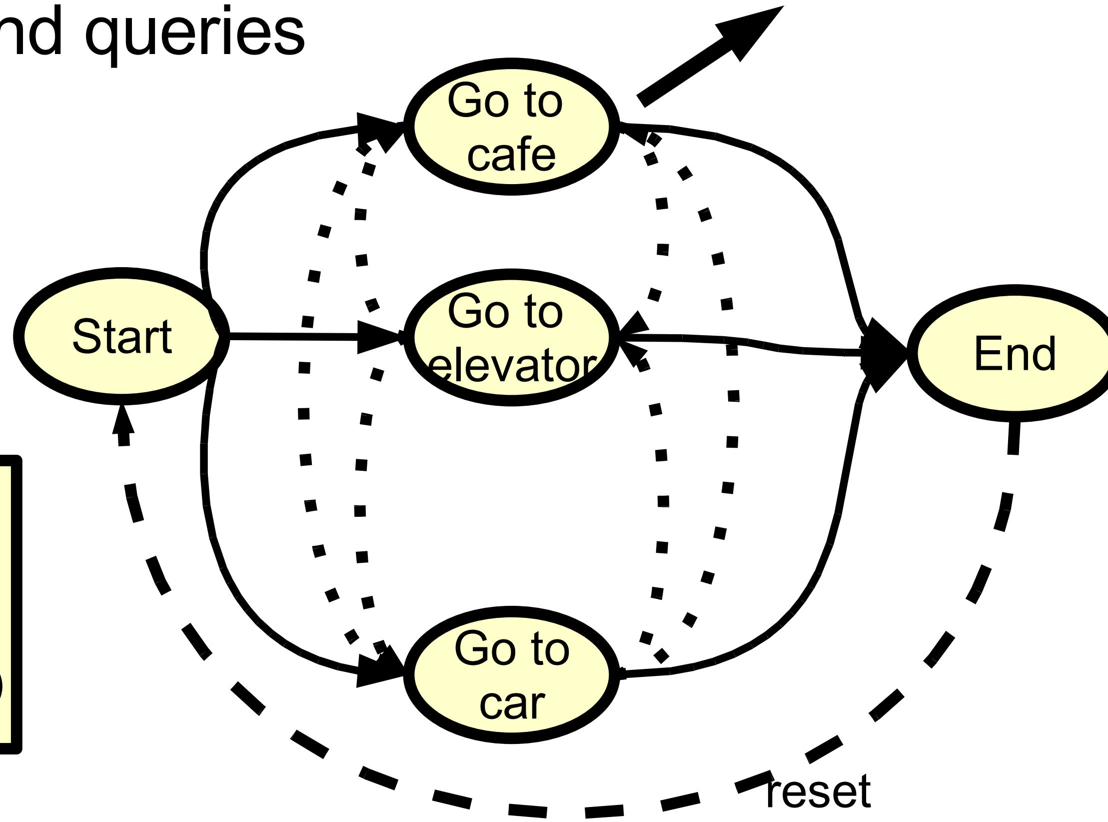
- Noisy speech recognition
  - ex. “Gates” becomes “Good”
- Linguistic ambiguities
  - multiple “elevators” may exist
  - the robot must know that an “elevator” is a location



CSAIL

The POMDP Dialog Model

- States (hidden!): the user's wants
- Observations: what the robot hears
- Actions: robot movements and queries
- Reward Model R(s,a)
- Transition Model T(s'|s,a)
- Observation Model O(o|s,a)



How do we specify the parameters?
(Our simple model has **1344** parameters!)

How can we learn the model online?

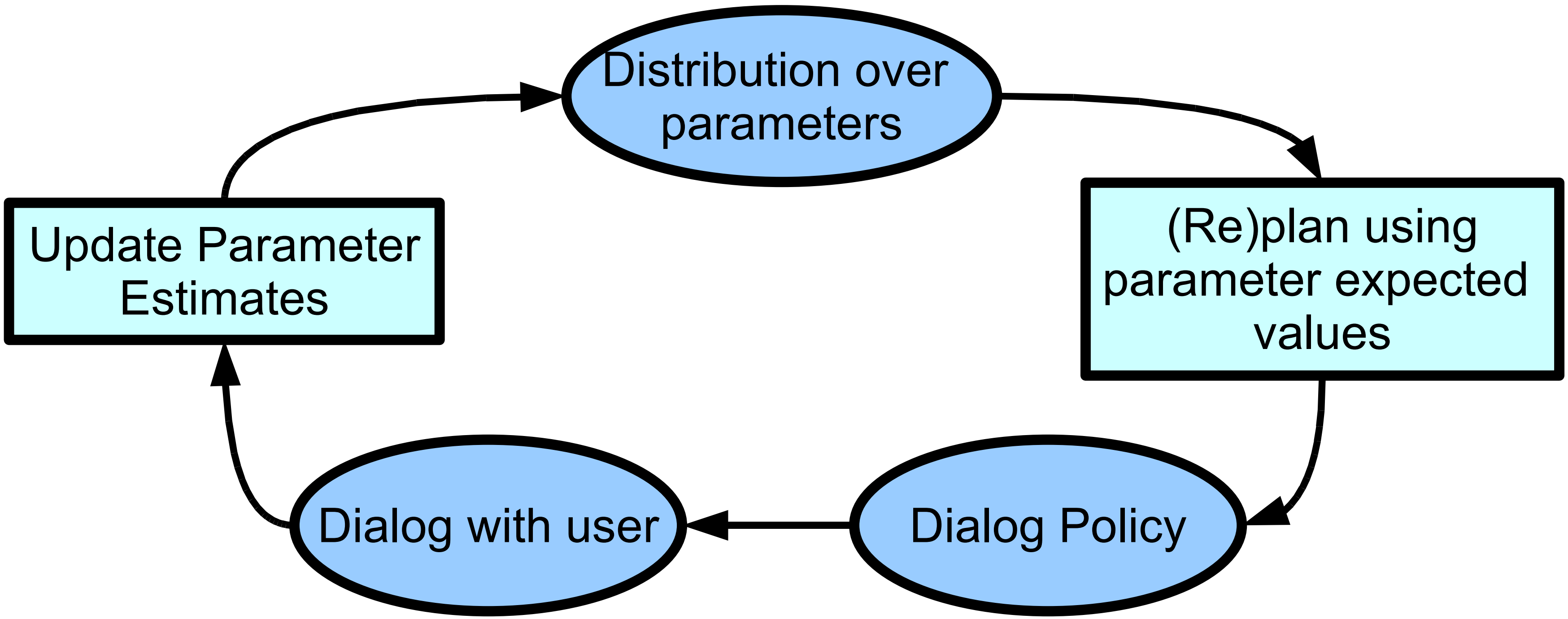
- We'll present several approaches that trade off between speed, robustness, and ease of use.
- Common theme:
  - Keep a distribution over possible models
  - Initialize that distribution with reasonable guesses about the parameters
  - Update our distribution (and our dialog policy) as we interact with the user

I. Planning with Expected Parameter Values

Idea: use only the expected values of the parameters

Benefit: very fast, works well in practice

Drawback: dialog manager unaware of uncertainty



Why is this reasonable?

Value of a belief

Value of belief, action pair

$$V_n(b) = max_a Q_n(b,a)$$

Future Reward

$$Q_n(b,a) = R(b,a) + \gamma \sum_{b' \in B} T(b'|b,a) V_{n-1}(b')$$

Expectation over states

Current reward

$$Q_n(b,a) = max_i q_{a,i} \cdot b$$

$$q_{a,i}(s) = E[R(s,a)] + \gamma \sum_{o \in O} \sum_{s' \in S} E[T(s'|s,a)] E[O(o|s',a)] V_{n-1,i}(s)$$

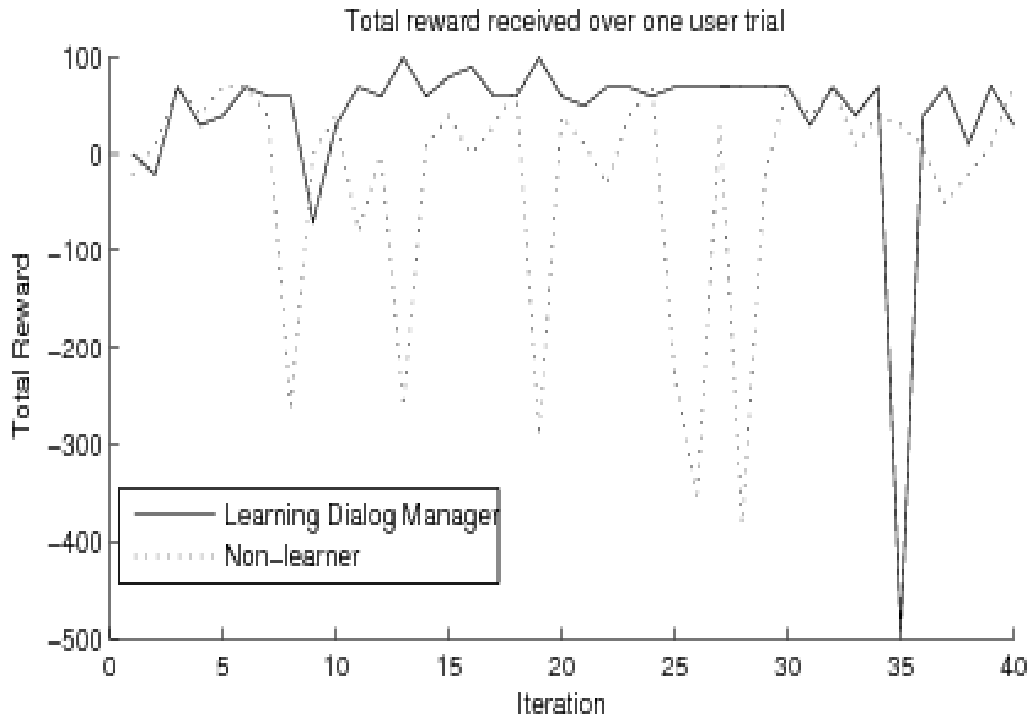
Expectations over model parameters

Expectation over model stochasticity

User Results

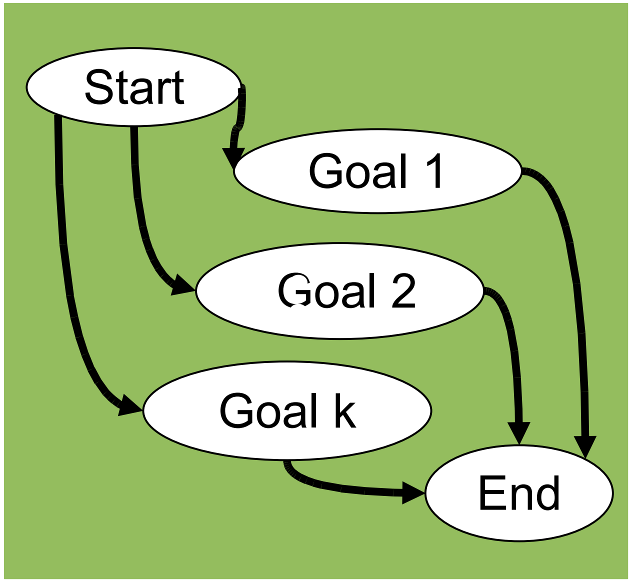
Non-learner
User: Take me to the elevator.
Robot: Where did you want to go?
User: The Gates elevator please.
Robot: Do you want to go to the Gates Tower?
User: Yes.
Robot: Going to Gates.

Learner
User: Take me to the elevator.
Robot: Do you want to go to the Gates Tower?
User: Yes
Robot: Going to Gates

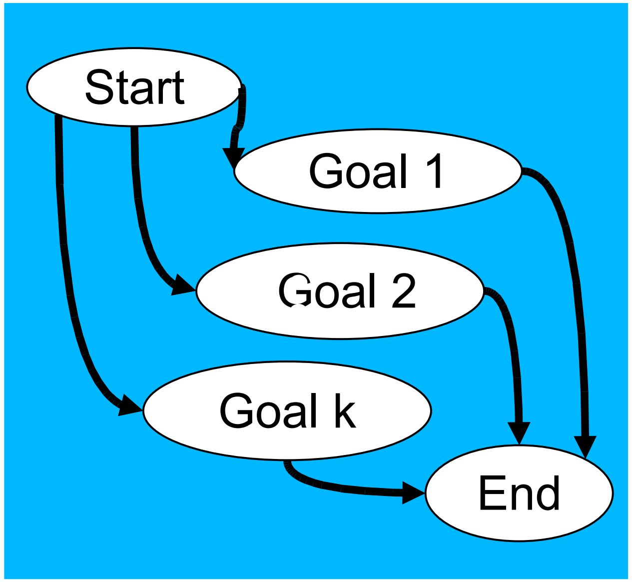


II. Parameters as Hidden State

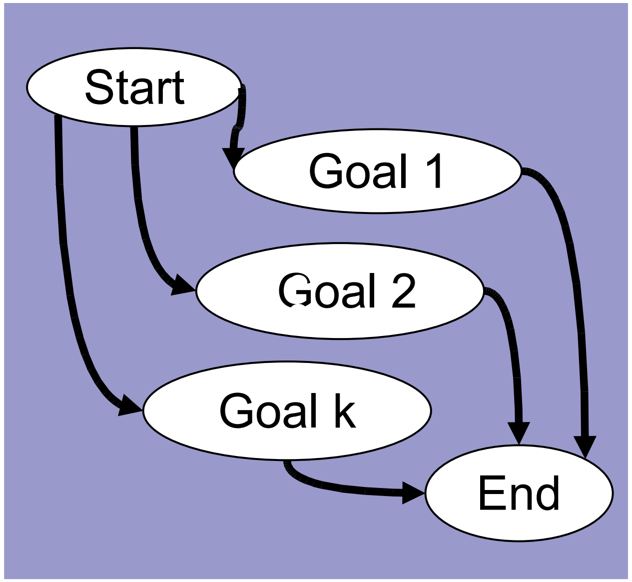
Idea: hidden state is now user's intent and preferences
Benefit: more robust dialog manager
Drawback: computationally very difficult, even with creative sampling to solve the big POMDP



User Model 1



User Model 2

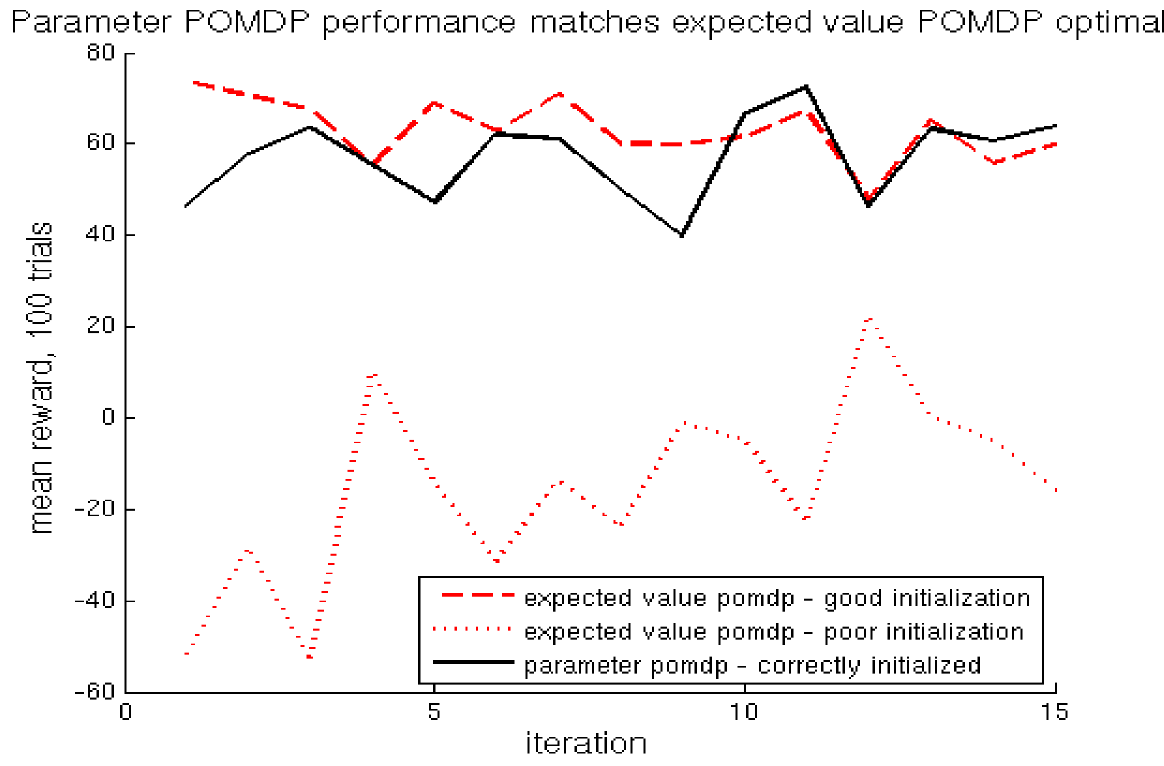


User Model 3

CSAIL

Simulation Results

The Parameter POMDP is not sensitive to the initial distributions placed over the parameters.



III. Meta-Action Queries

Idea: robot asks the user about what action to do next
Benefits:

- Overall less feedback required (robot determines when additional learning is needed)
- We can discover the consequences of a mistake without making the mistake.

Case One: Discrete User Models

- Choose sets of parameters that produce different policies; let each of these be a user preference model.
- Design meta-action queries to differentiate between the models.
- Solve just like the parameter POMDP.

CSAIL

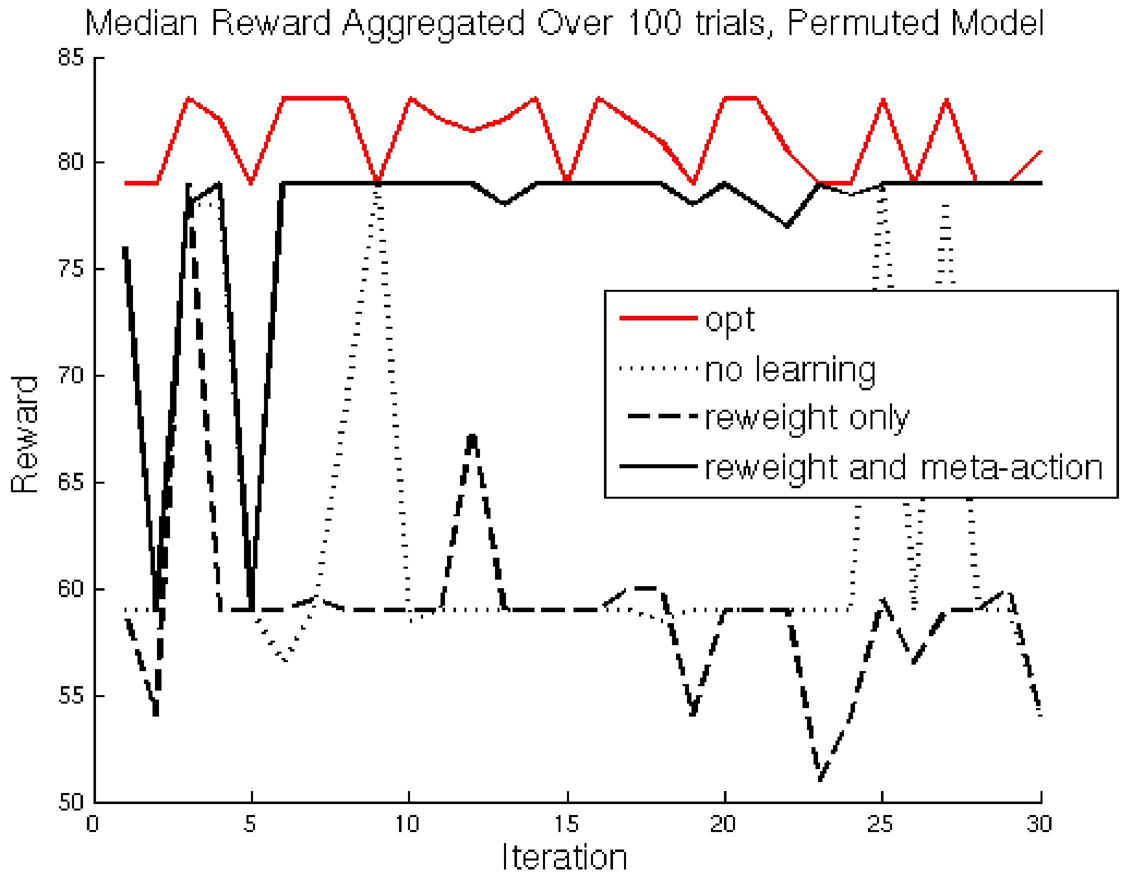
Case Two: Continuous User Models

- Sample many POMDPs from an initial distribution over observations and rewards.
- Combine the POMDP samples to choose actions:
  - Find the action with the minimal risk:

$$a = argmin_{a \in A} \sum_i \left(Q_i(b, a) - Q_i(b, a') \right) w_i$$

  - If the risk is more than the cost of a meta-action, check if the meta-action will reduce the risk.
- Reweight and resample POMDPs as new data arrives and our distribution over models changes.

Case Two: Continuous Models



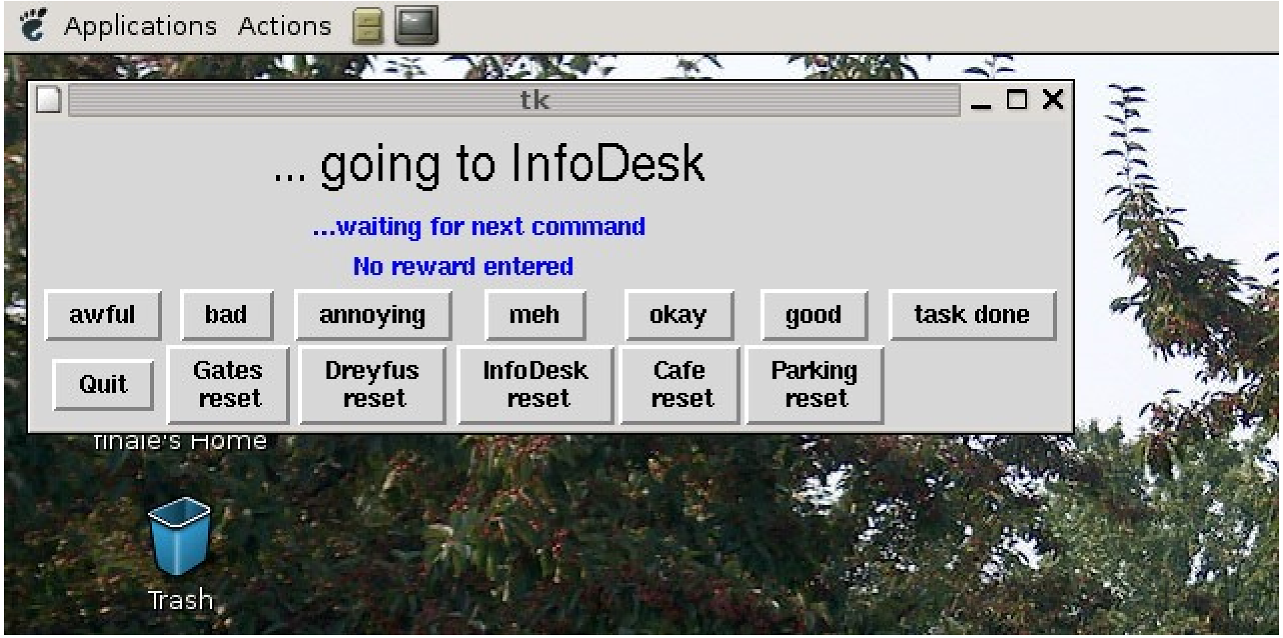
Wheelchair Video

Conclusions and Future Work

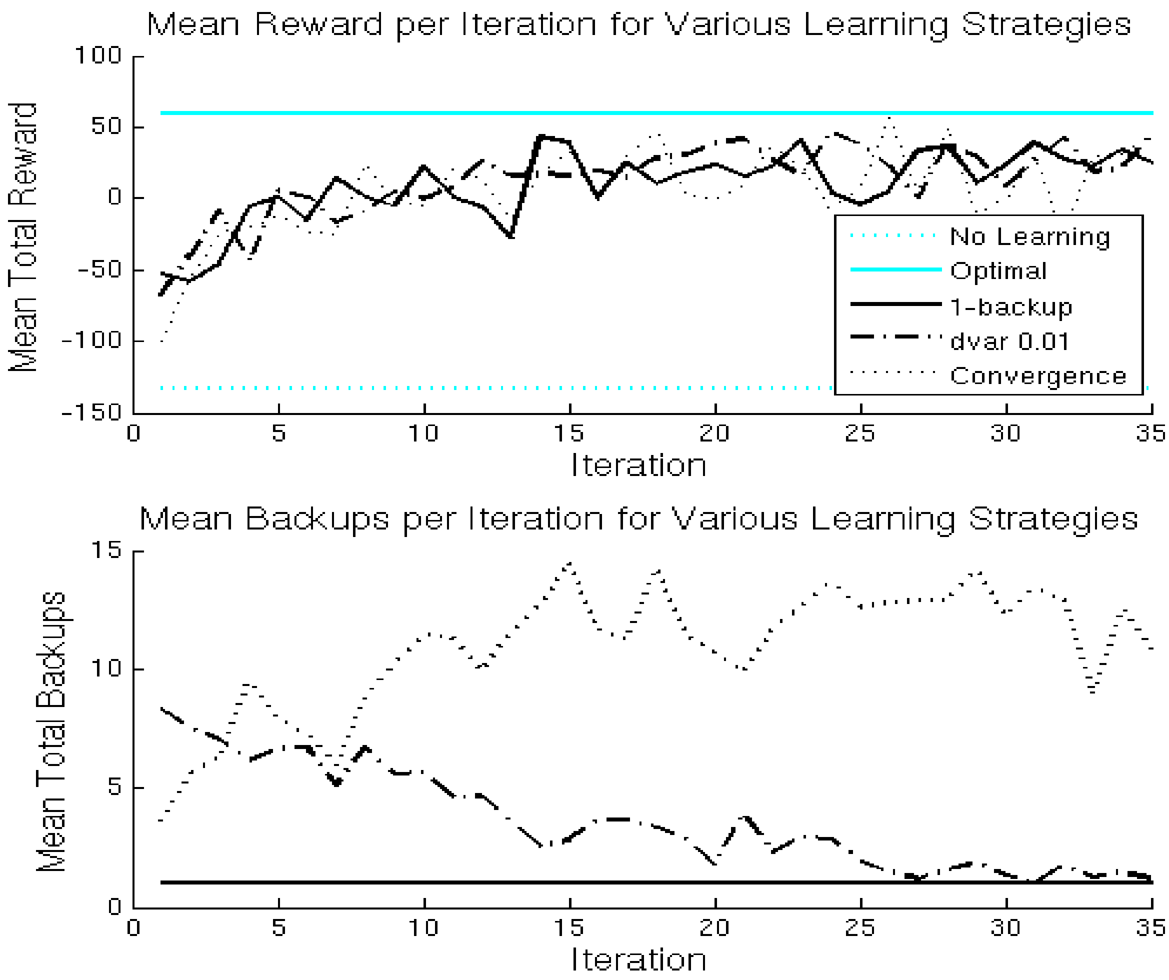
- POMDPs provide a useful way to handle dialog uncertainty.
- Although the dialog models require many parameters, we presented several approaches for learning those parameters online.
- The learning process can be further improved by incorporating meta-actions, actions that ask about what the robot should have done.

Thank-you

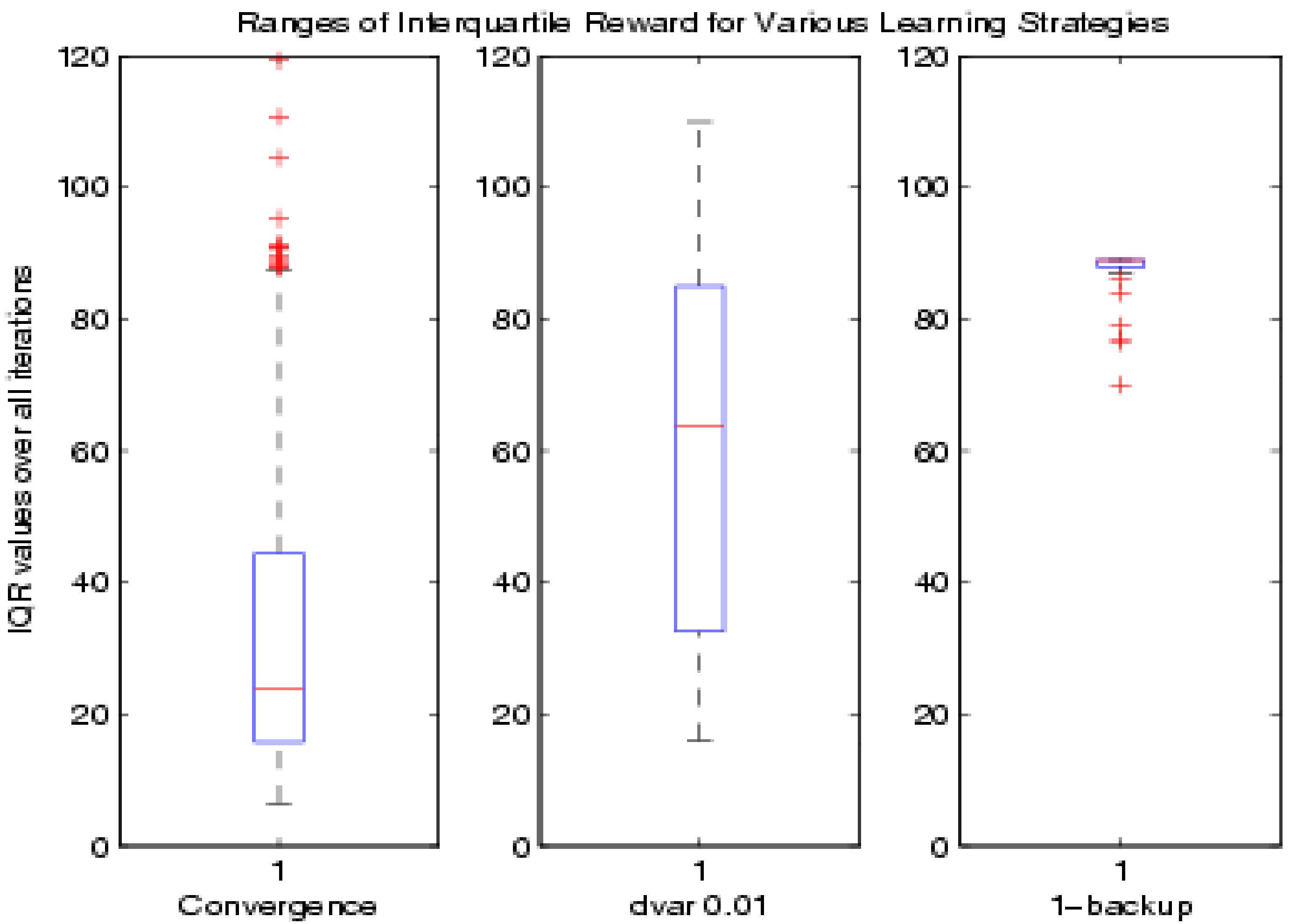
User Interface



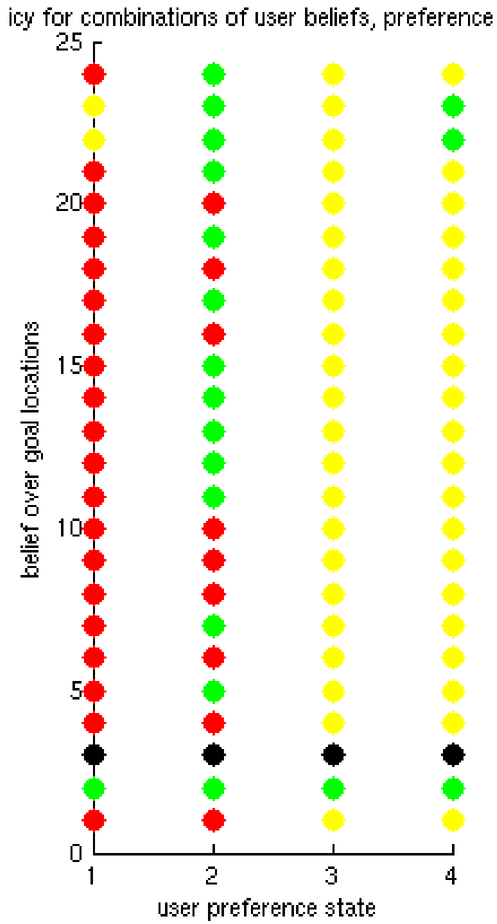
Expected Value Simulation Results



Expected Value Simulation Results



Dialogs have limited policies



CSAIL

Meta-Actions help prune rewards