Learning User Models with Limited Reinforcement: An Adaptive Human-Robot Interaction System

Finale Doshi
Nick Roy

Motivation
(our goals, how you can help)

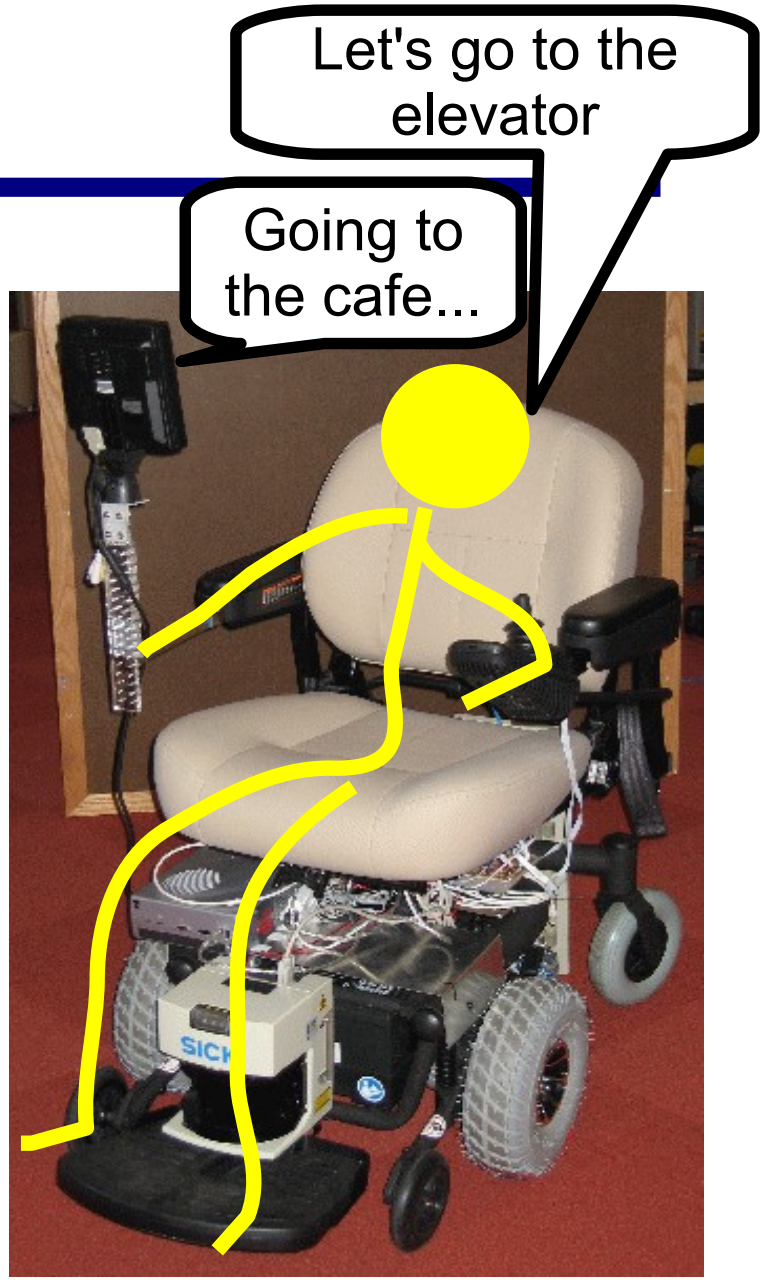
Motivation

- Wouldn't it be nice if we could simply tell a robot to go to a particular location? Follow along side someone?
- This ability would be particularly useful to wheelchair users with severely limited mobility.

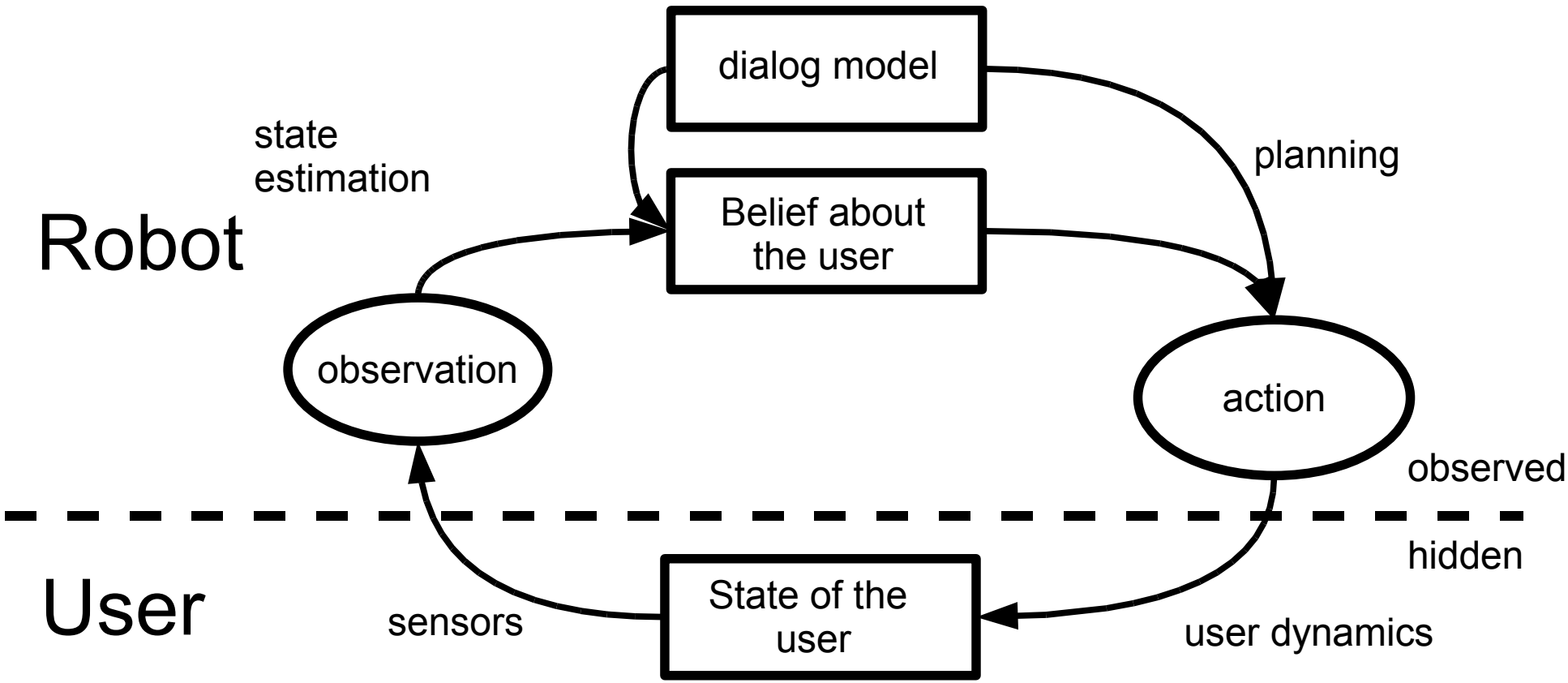
The Problem

Speaking to robots involves several challenges:

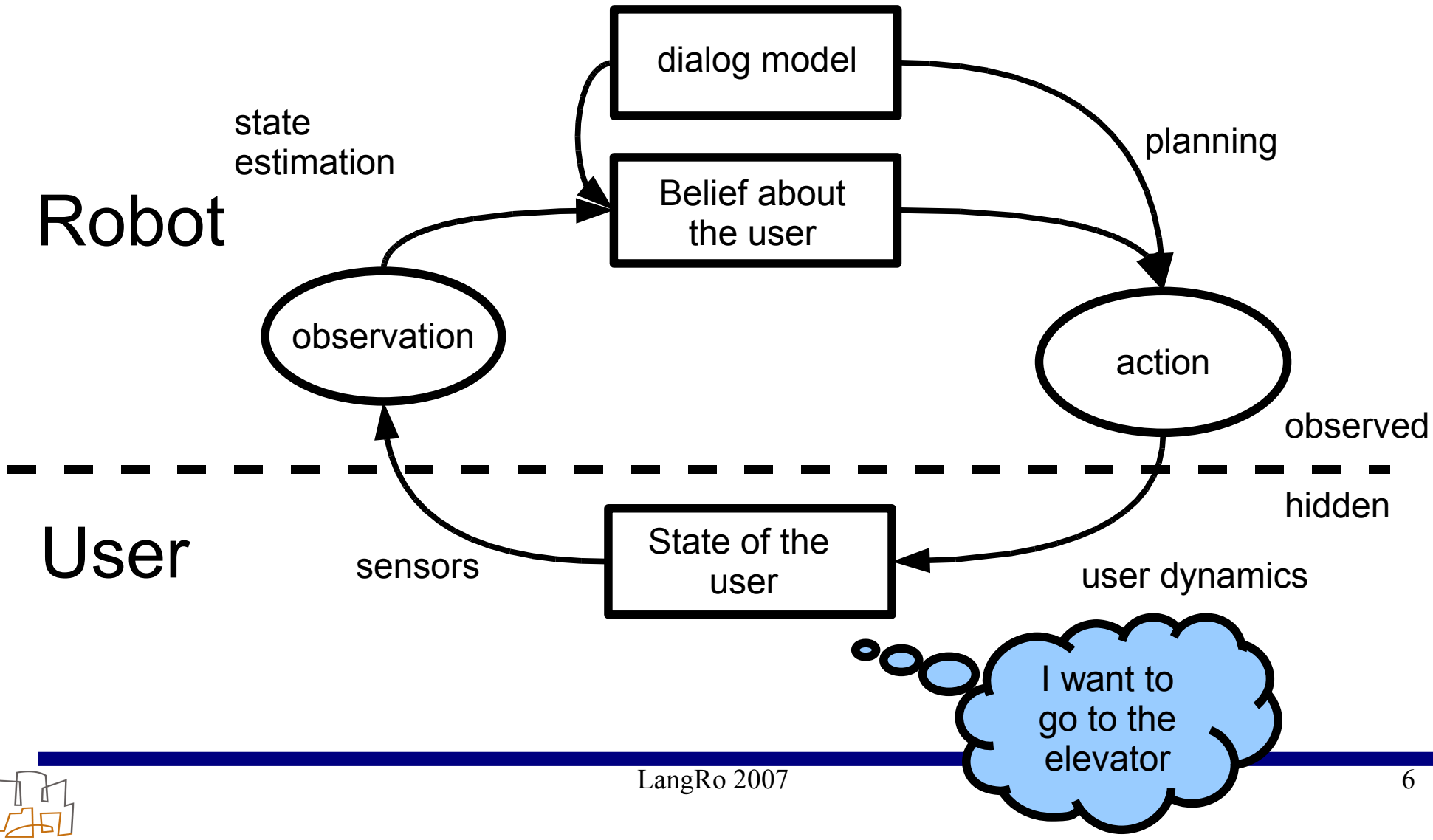
- Noisy speech recognition
- Linguistic ambiguities
  - multiple “elevators” may exist
  - the robot must know that an “elevator” is a location
  - other ambiguous phrases



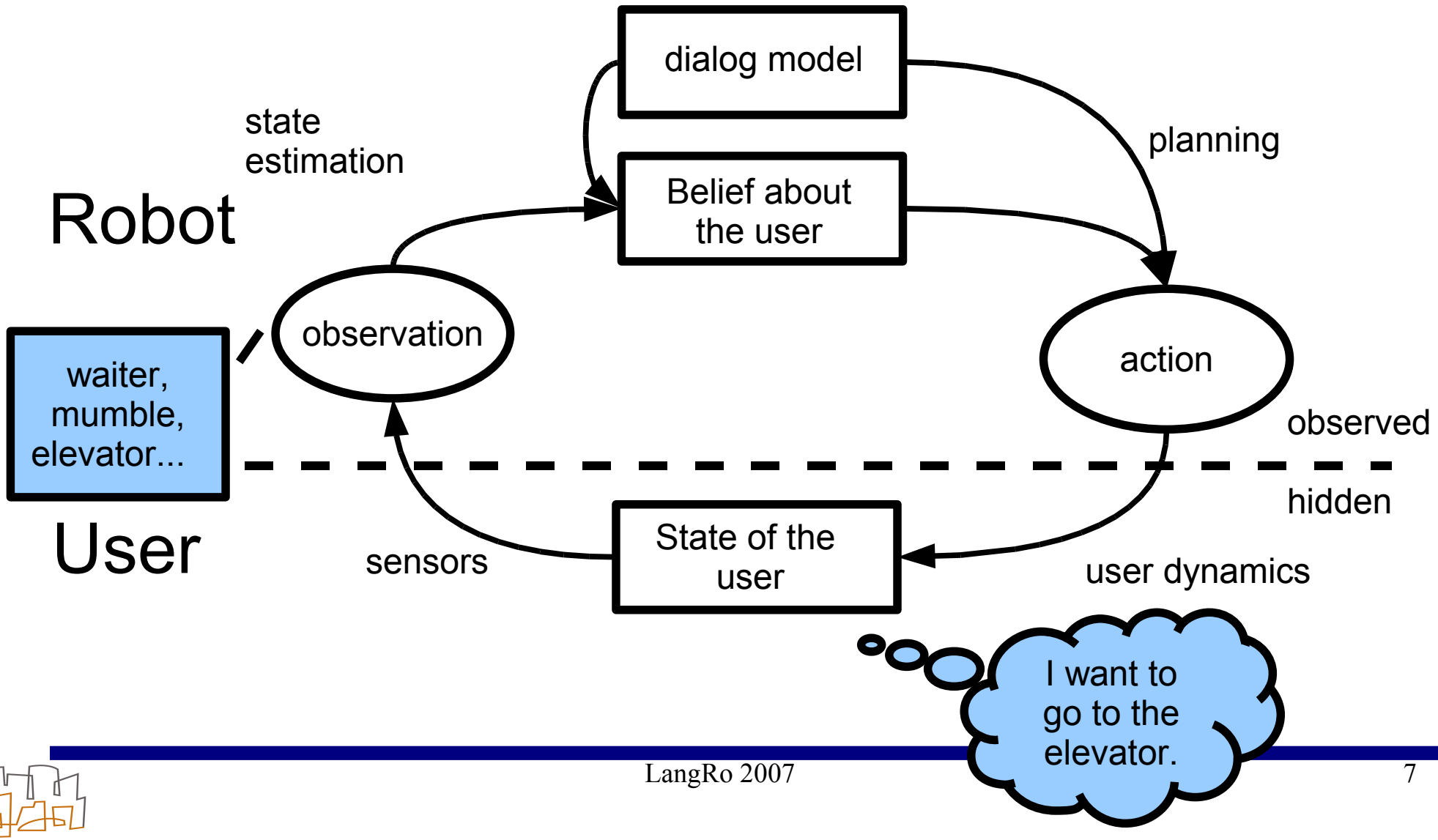
The planning process



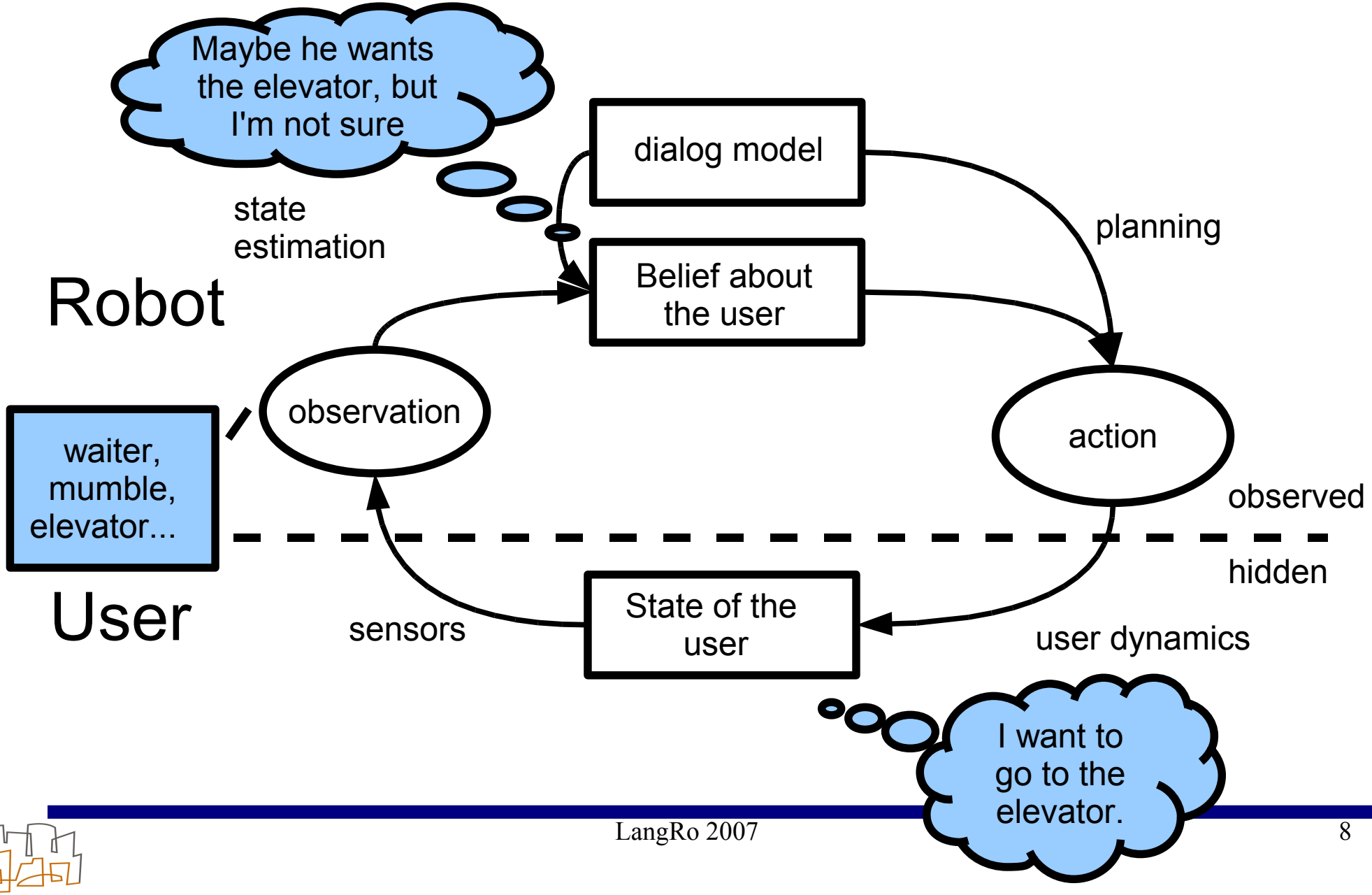
The planning process



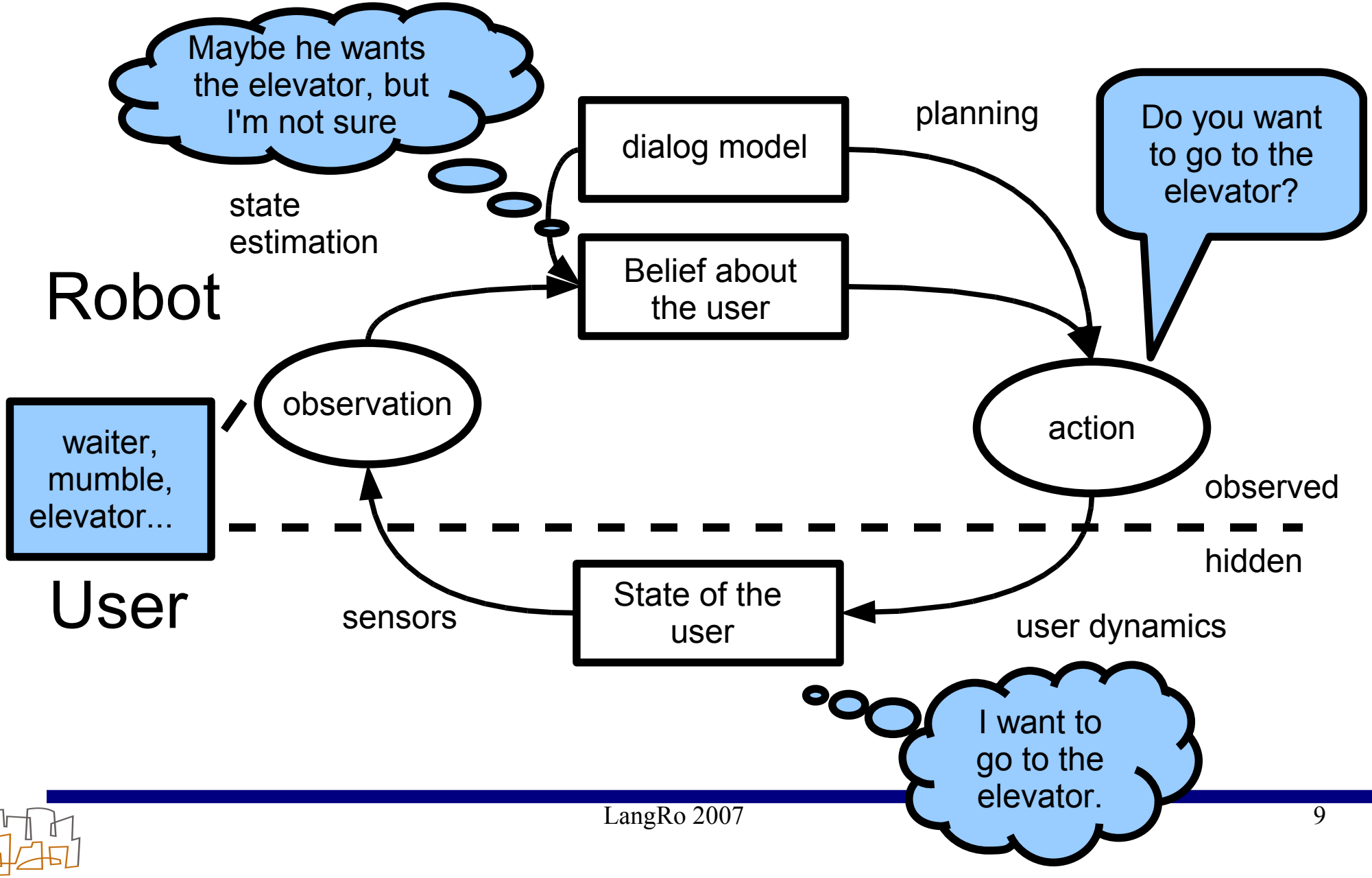
The planning process



The planning process



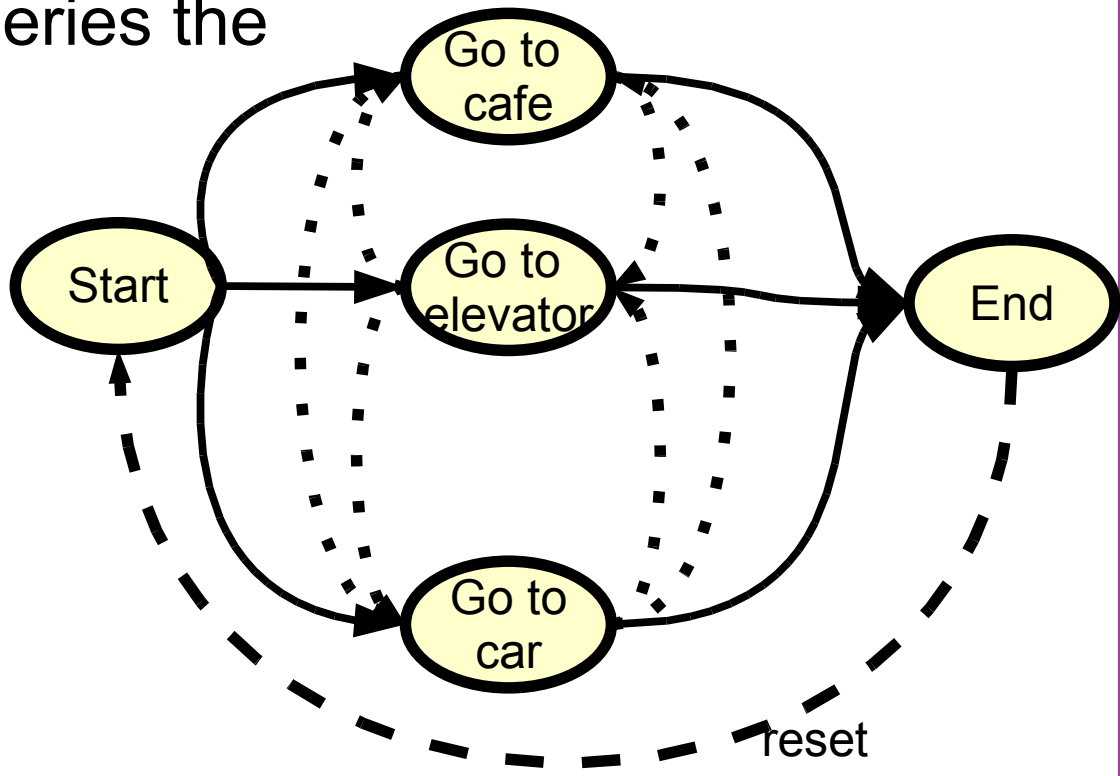
The planning process



The POMDP Dialog Model

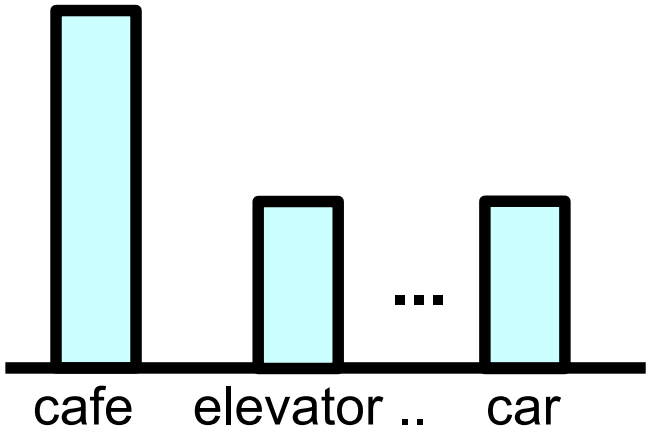
Partially Observable Markov Decision Process

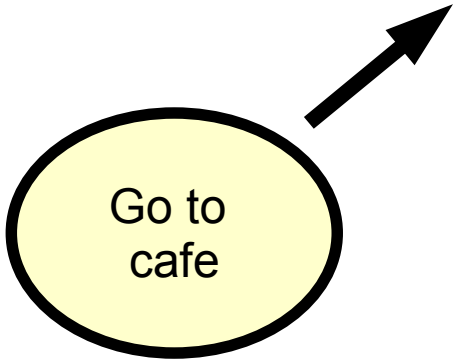
- States (hidden!): the user's wants
- Observations: what the robot hears
- Actions: movements and queries the robot can do
- Reward Model R(s,a)
- Transition Model T(s'|s,a)
- Observation Model O(o|s,a)



Observation Model

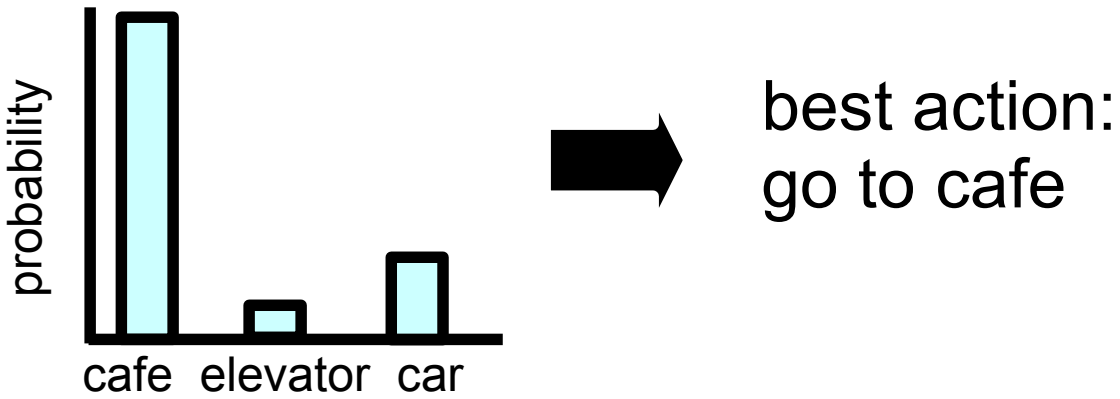
- Currently we use a bag of words approach with keyword list.
  - Each location has a keyword (e.g., 'cafe' for cafe)
  - Additional keywords also included (e.g., 'food', 'tea', or 'lunch')
- Model is probability of each observation in each state.
- Could be adjusted to handle more sophisticated language models.





Why is the POMDP model useful?

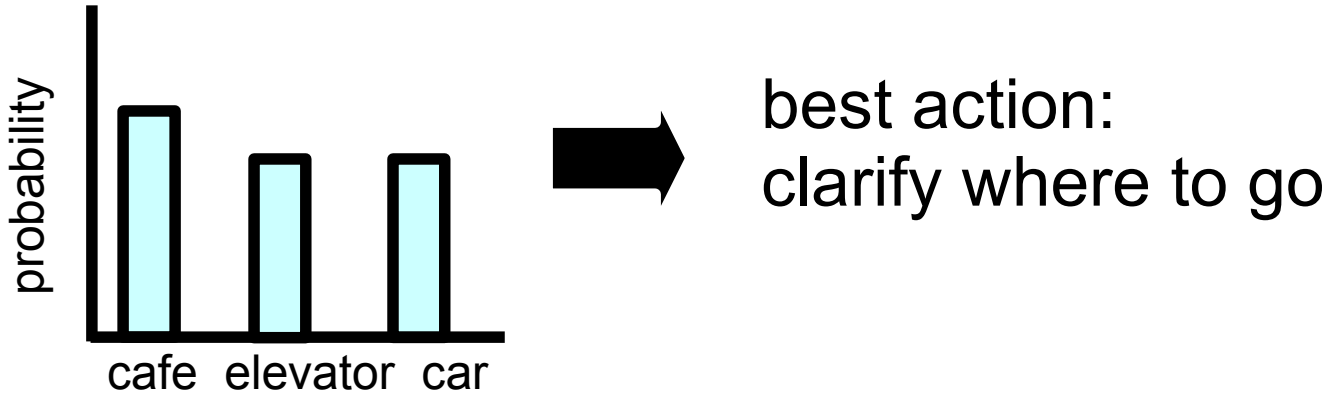
- We track our *belief*, a probability distribution over states.
- We choose an action based on our belief, thus taking into account our uncertainty about what the user really wants.



- POMDPs have been used in several dialog management applications, such as Roy, Pineau, and Thrun (2000) and Williams and Young (2005)

Why is the POMDP model useful?

- We track our *belief*, a probability distribution over states.
- We choose an action based on our belief, thus taking into account our uncertainty about what the user really wants.



- POMDPs have been used in several dialog management applications, such as Roy, Pineau, and Thrun (2000) and Williams and Young (2005)

Difficulties with the POMDP model

R(user wants cafe, go to cafe) R(user wants cafe, go to coffee machine) R(user wants cafe, go to printer) R(user wants cafe, go to cafe) R(user wants cafe, go to copy machine) O(hear cafe|user wants cafe, ask where) O(hear cafe|user wants coffee machine, ask where) O(hear cafe|user wants cafe, ask where) O(hear cafe|user wants printer, ask where) O(hear cafe|user wants copy machine, ask where) O(hear c e|user wants coffee machine, confirr cafe) O(hear cafe| user wants copy ts printer, confirm cafe) O(hear ca (hear cafe|user wants coffee ma ser wants cafe, confirm coffee m e, confirm coffee machine) O(hea ine) O(hear cafe| user wants cafe e machine, confirm printer) O(hear cafe|user wants copy machine, confirm printer) O(hear cafe|user wants printer, confirm printer) O(hear cafe|user wants cafe, confirm copy machine) O(hear cafe|user wants coffee machine, confirm copy machine) O(hear cafe|user wants cafe, confirm copy machine) O(hear cafe|user wants copy machine, confirm copy machine) O(hear cafe|user wants printer, confirm copy machine) O(hear cafe|user wants

Even a 5-state model has **1344** parameters!

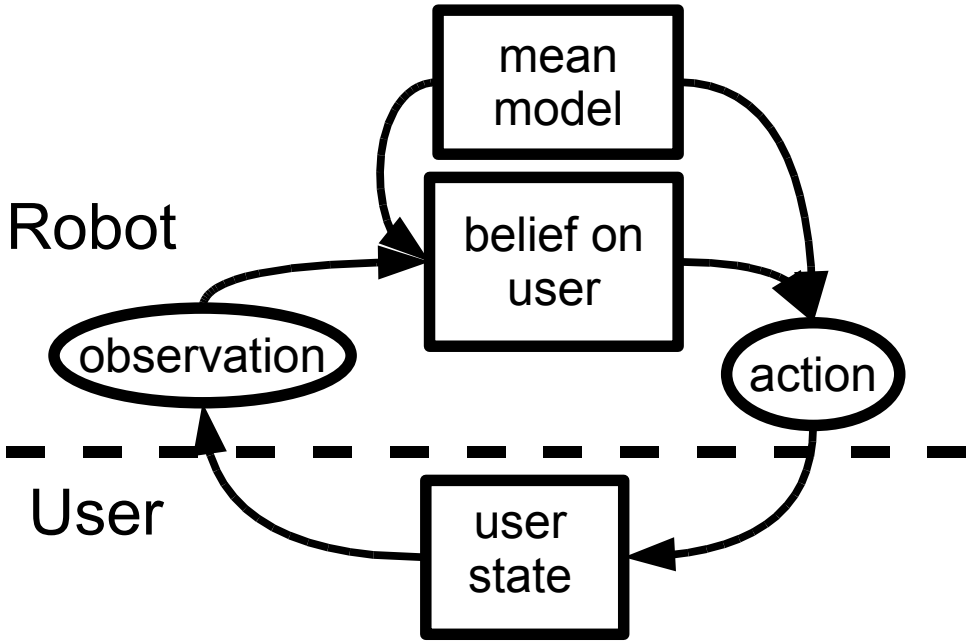
We would like to learn these parameters online.

CSAIL

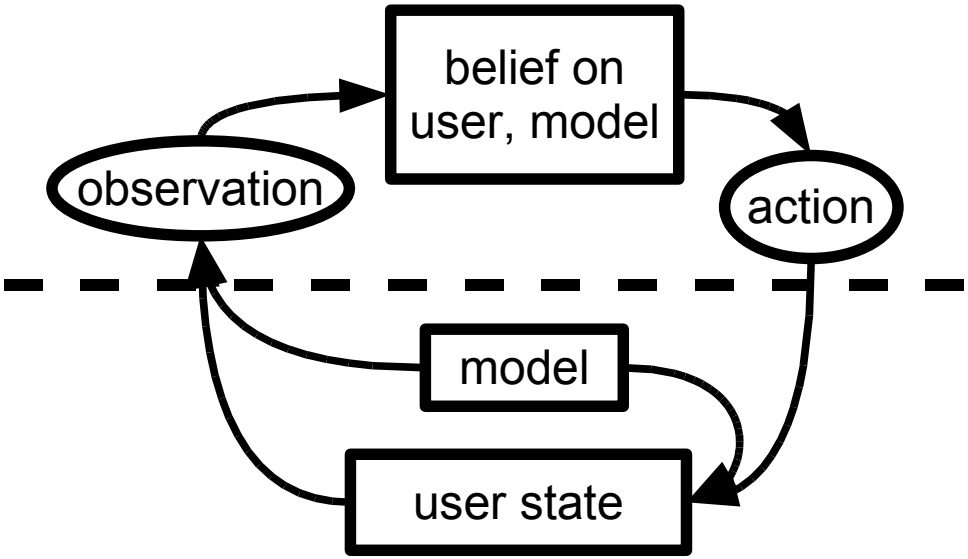
Algorithm
(learning the user model)

How can we learn the model online?

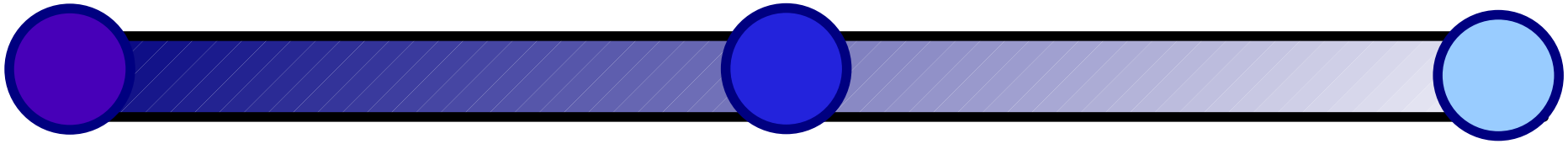
Ignore uncertainty:
fast, not robust



Plan with parameters as hidden state:
robust but slow



How can we learn the model online?



Ignore uncertainty:
fast, not robust

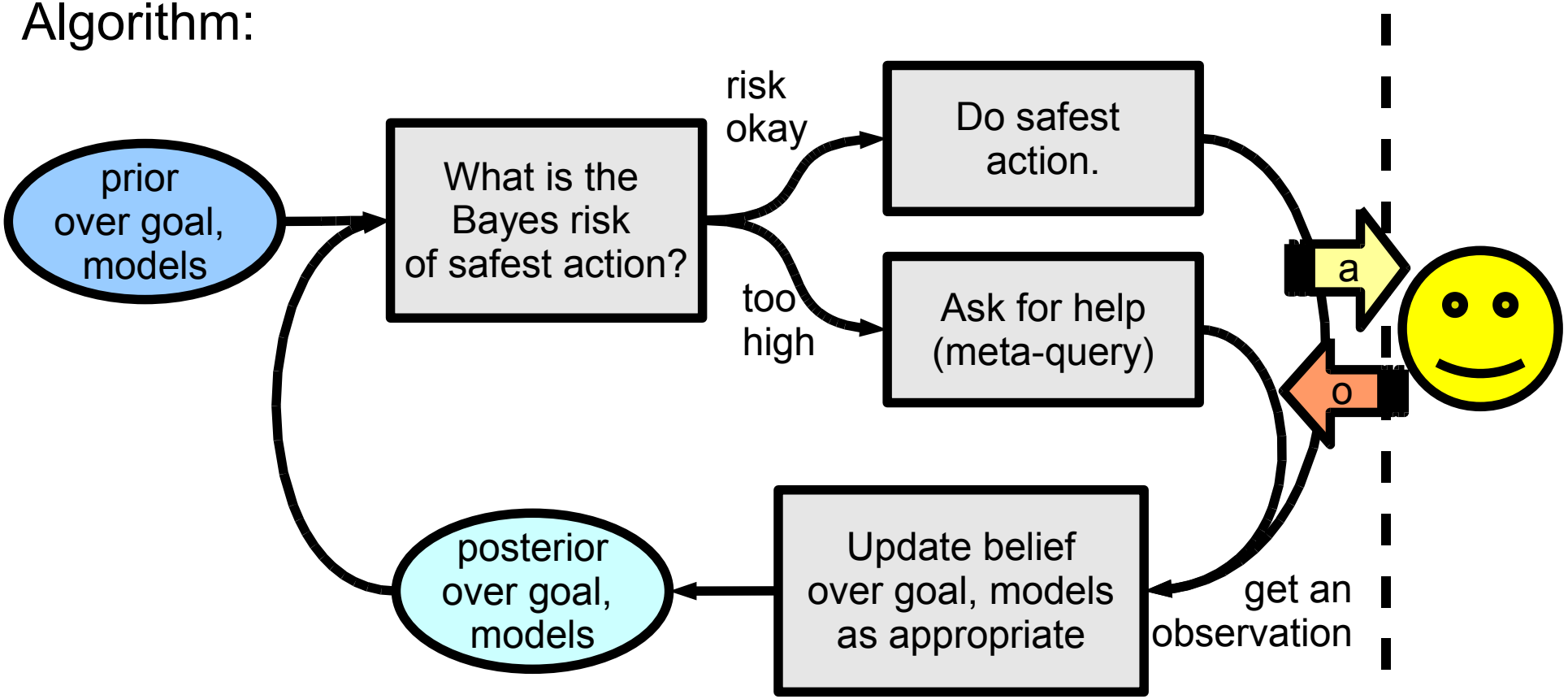
Approximate planning with Bayes risk, meta-queries

Plan with parameters as hidden state:
robust but slow

Approximate Planning Strategy

Idea: approximate risk of each action; if risk is large, ask for help

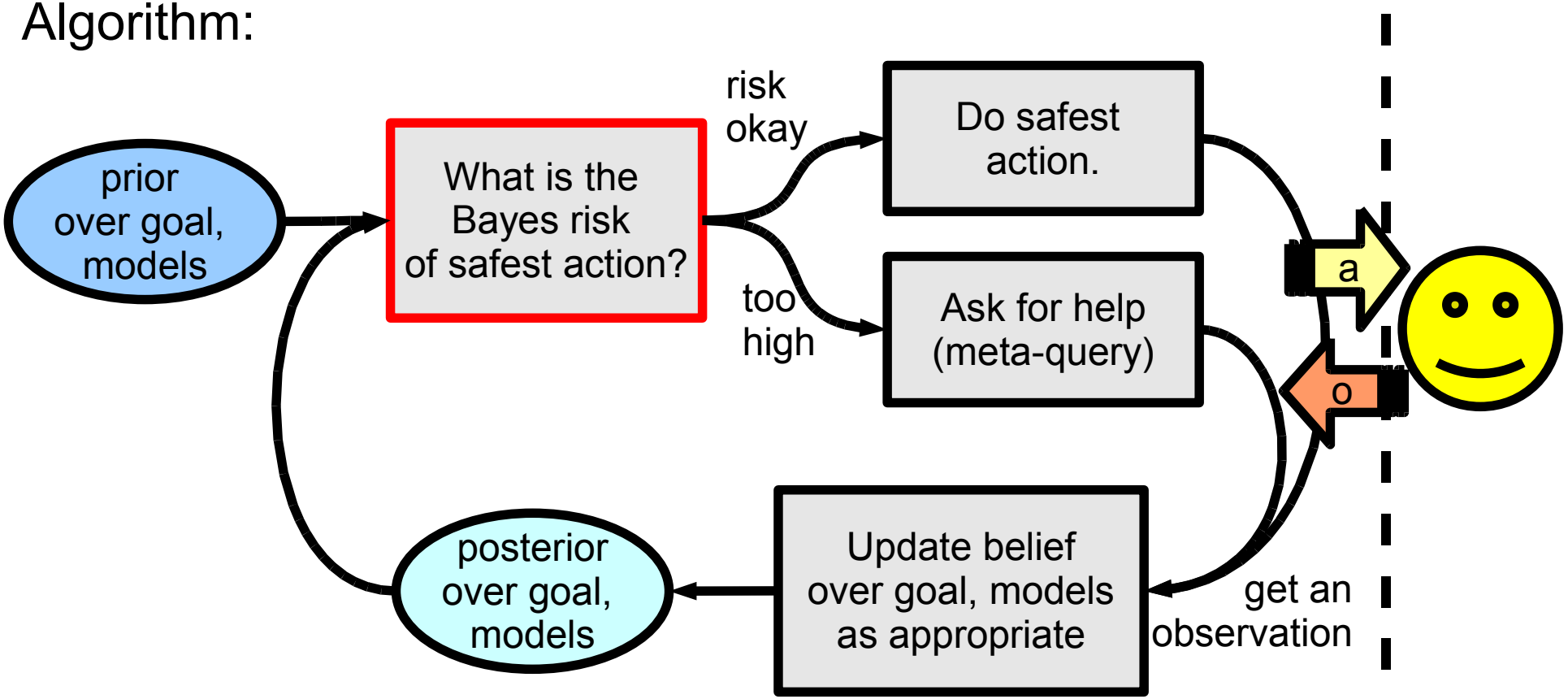
Algorithm:



Approximate Planning Strategy

Idea: approximate risk of each action; if risk is large, ask for help

Algorithm:



Action Selection with Bayes Risk

- Find the action with the minimal risk:

$$a = argmin_{a \in A} \int_M (Q_m(b_m, a) - Q_m(b_m, a_m')) p(m) dm$$

If the risk is more than the cost of a meta-query, ask for help.

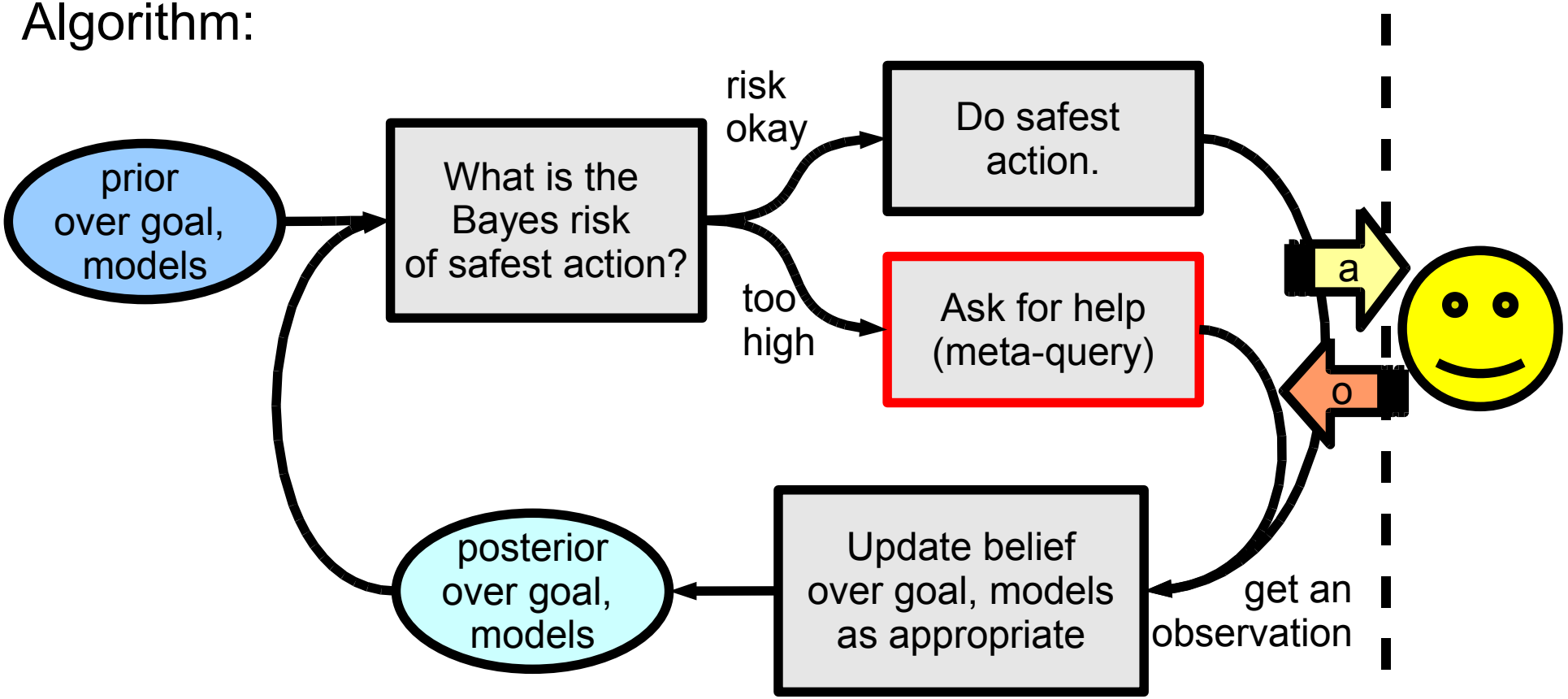
- Evaluate the Bayes Risk integral approximately using sampled P0MDPs:

$$a = argmin_{a \in A} \sum_i (Q_i(b_i, a) - Q_i(b_i, a_i')) w_i$$

Approximate Planning Strategy

Idea: approximate risk of each action; if risk is large, ask for help

Algorithm:



Meta-Queries

Idea: give the user a sense of the robot's uncertainty so that he can provide feedback to help discover the model.

Benefits:

- Robot does not need to take large risks to determine that a particular decision may be poor.
- User only needs to provide reinforcement when the robot is sufficiently confused.

Meta-Queries

Questions of the form:

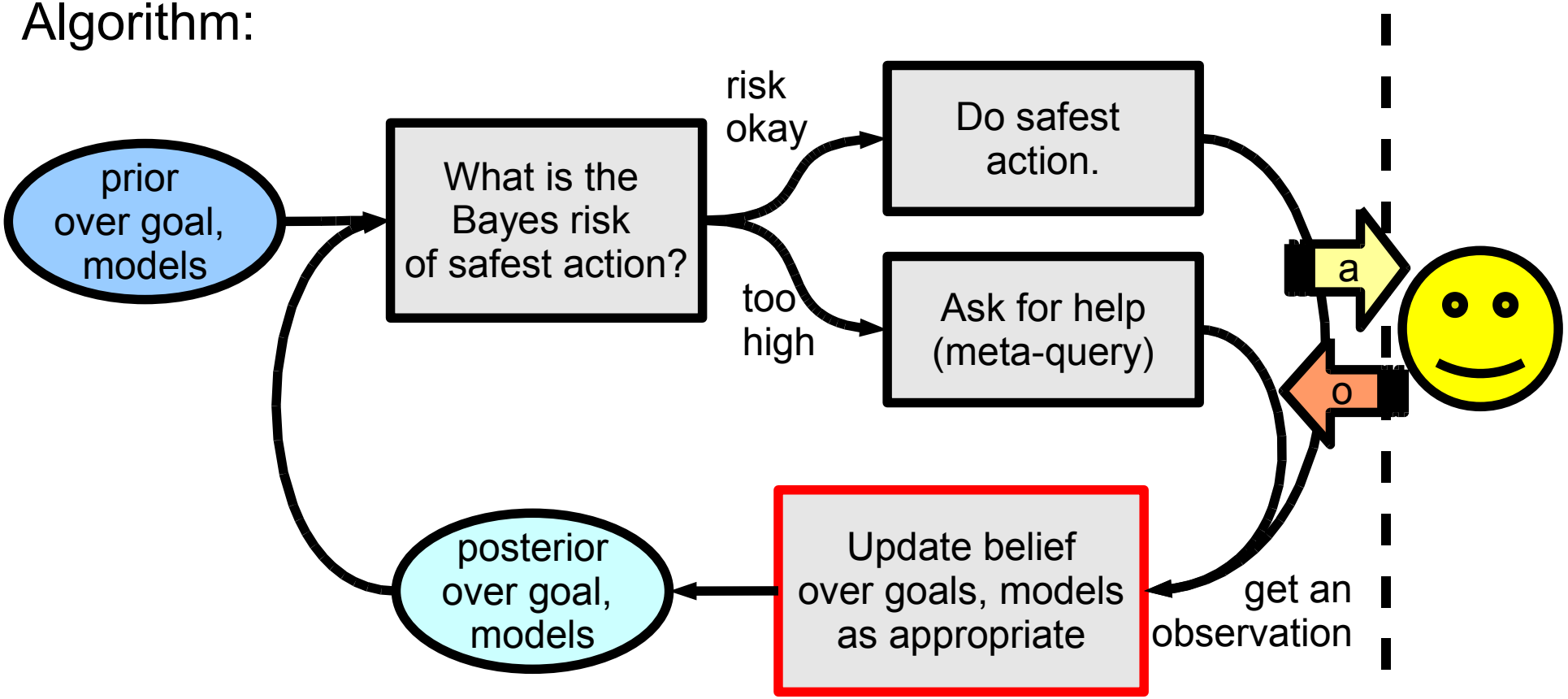
- “I think you **might** want to go to the printer. Should I go to the printer?”
- “I'm **certain** you want to go to the printer. Should I go to the printer?”
- “Instead, should I ask for you to confirm your location?”

Ask these questions until we determine the correct action.

Approximate Planning Strategy

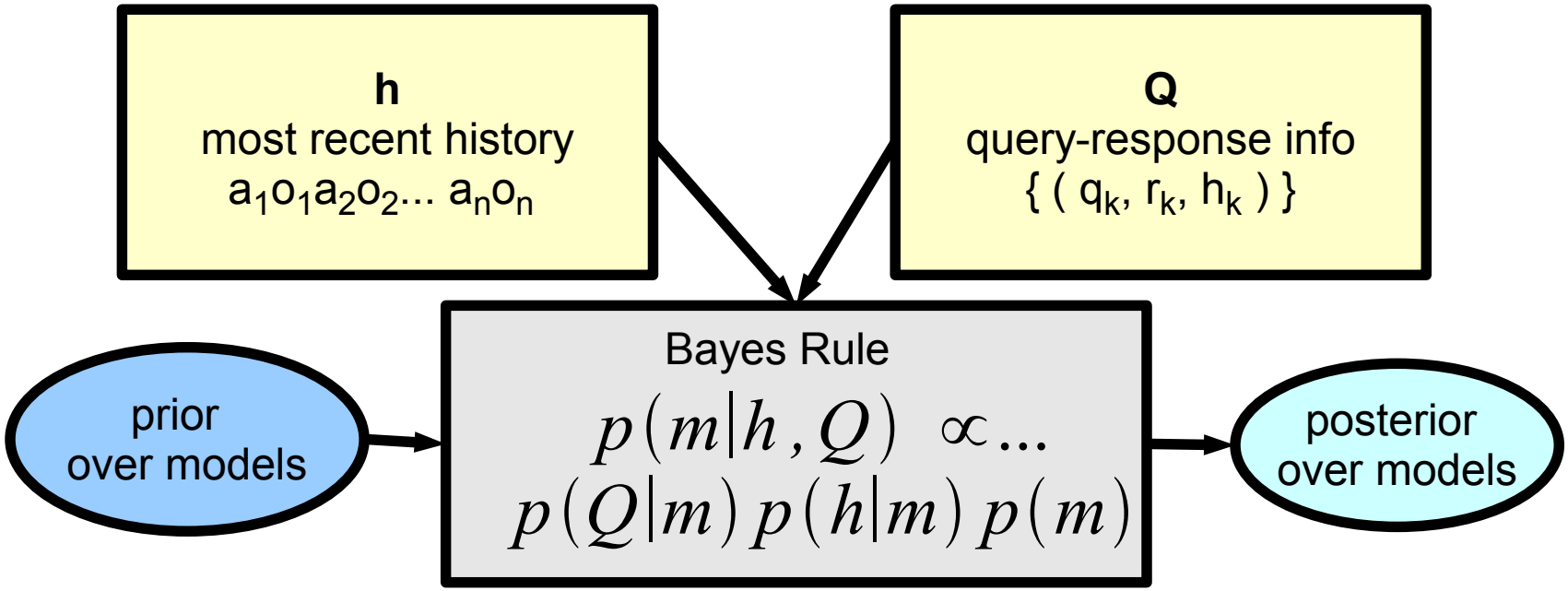
Idea: approximate risk of each action; if risk is large, ask for help

Algorithm:



How to Sample POMDPs: Belief Update

Two sources of information affect our belief over models:

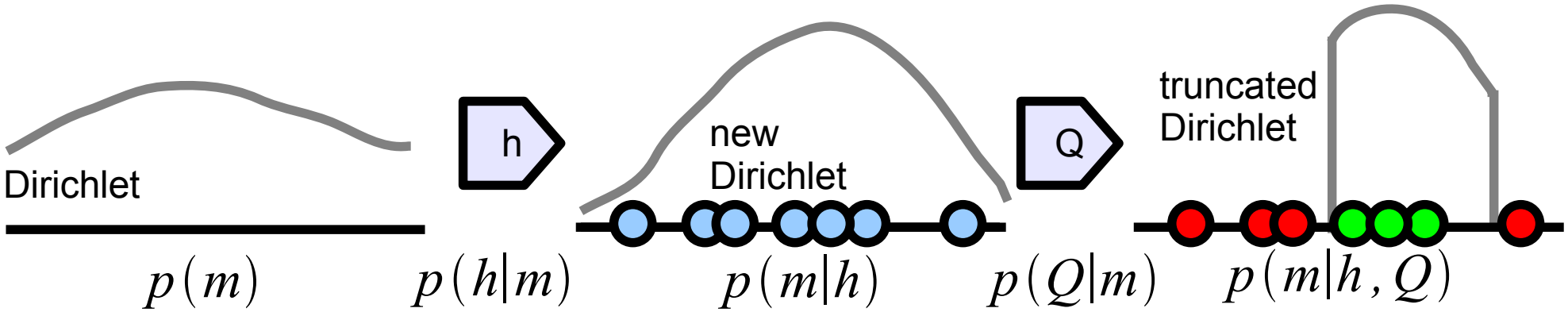


Note: we resample when update is likely to be accurate (e.g., at the completion of a dialog).

CSAIL

Belief Update

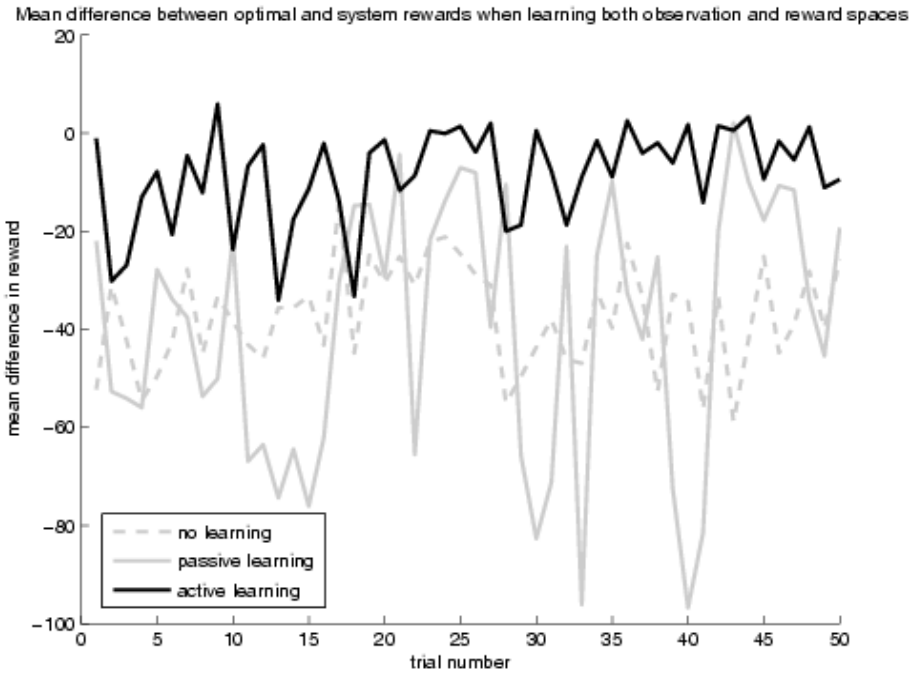
- If we begin with a Dirichlet prior, the history information can be incorporated in closed form.
- We sample from the this updated distribution over models, throwing out samples that violate too many meta-queries.

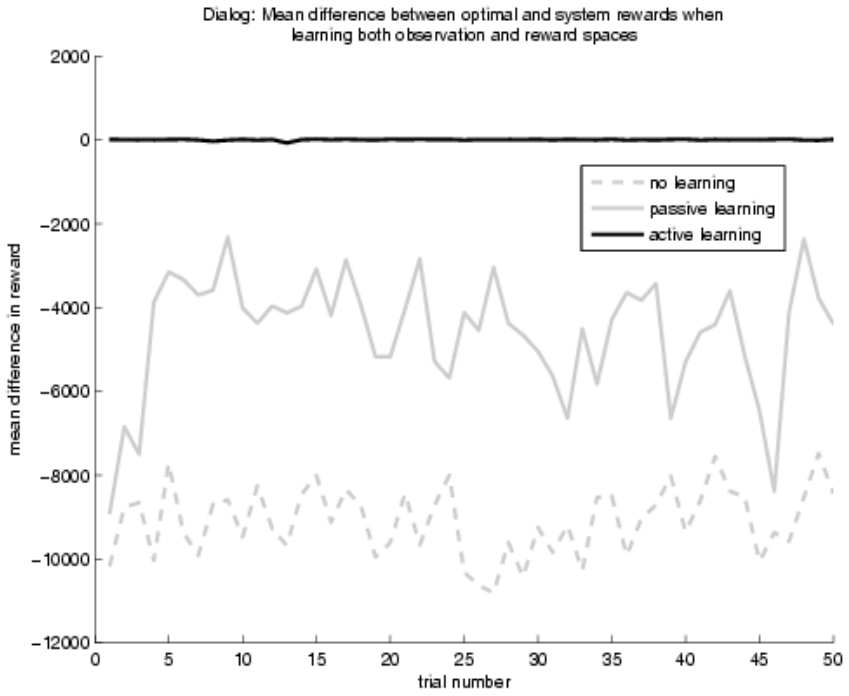


(Note: we require rejection sampling because we cannot compute how the Dirichlet is truncated.)

Results

Simulation Results: Full Model Learning





Reasonable Prior

Non-informative Prior

CSAIL

User Test Results: Example Dialog 1

Early Conversation:
User: Give me the forecast.
Robot: I'm confused. What action should I take now?
<User indicates that the robot should provide the weather forecast>
Robot: Showers

Later Conversation:
User: What's the forecast for today?
Robot: Do you want the weather?
User: Yup.
Robot: Showers

User Test Results: Example Dialog 2

Early Conversation:
User: I want some tea from the kitchen.
Robot: How can I help you?
User: The tea is in the kitchen near the coffee machine. <*The system hears ``copy'' instead of ``coffee'' and is confused about “tea” and “kitchen.''* >
Robot: How can I help you?
User: Go to the kitchen.
Robot: I'm confused. What action should I take now?
<*The user indicates ``Confirm the coffee machine area'' is the correct action.* >
Robot: Do you want to go to the coffee machine area?
User: Yup.
...
<*The system asks a few more (non-meta) questions*>
Robot: Do you want to go to the coffee machine area?
User: Yup.
Robot: Going to the coffee machine area.

Later Conversation:
User: Take me to the kitchen. <*The system hears “Nick” and ``kitchen.''*>
Robot: How can I help you?
User: I want some coffee. <*The system hears ``copy'' instead of ``coffee.''*>
Robot: Do you want to go to the coffee machine area?
User: Yup.
Robot: Going to the coffee machine area.

Conclusions and Future Work

- Although POMDP dialog models require many parameters, we can learn those parameters online.
  - Bayes risk action selection lets the robot act robustly in the face of model uncertainty
  - The learning process can be further improved by incorporating meta-queries.
- Extensions
  - Improve efficiency (better sampling of POMDPs)
  - **More intuitive meta-query mechanisms**
  - Coupling with **more sophisticated language models**

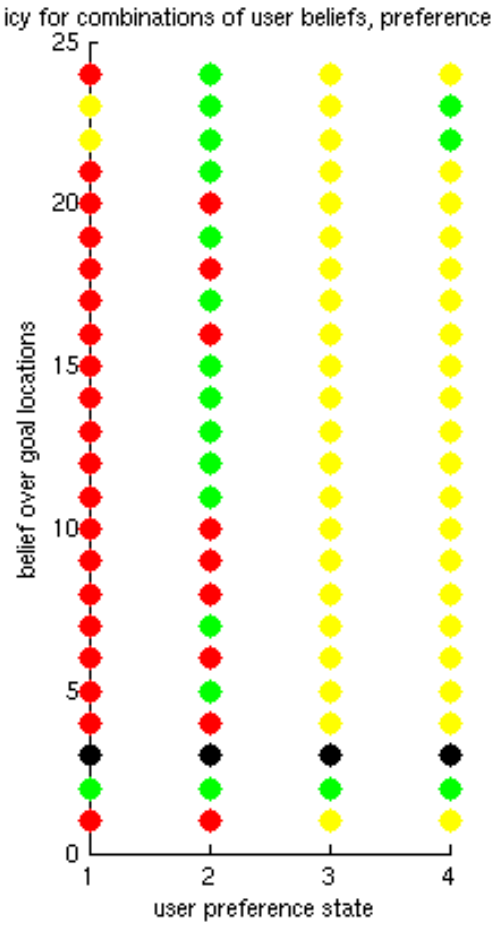
Thank-you!

Discrete Models: Why few policies?

In the special case where:

- Only rewards are unknown
- Simple dialog model

The *policies* for a variety of parameter values are similar; the main degree of freedom is how certain we must be before acting, which translates to how many times to confirm a choice.



Error in Approximating Bayes Risk

- If we want to estimate if the Bayes Risk is greater than ζ with confidence δ, two error sources exist:
  - Error due to approximating risk from samples:

$$n_m = \frac{(R_{max} - min(\zeta, R_{min}))^2}{2(1-\gamma)^2 \epsilon_m^2} \log \frac{1}{\delta}$$

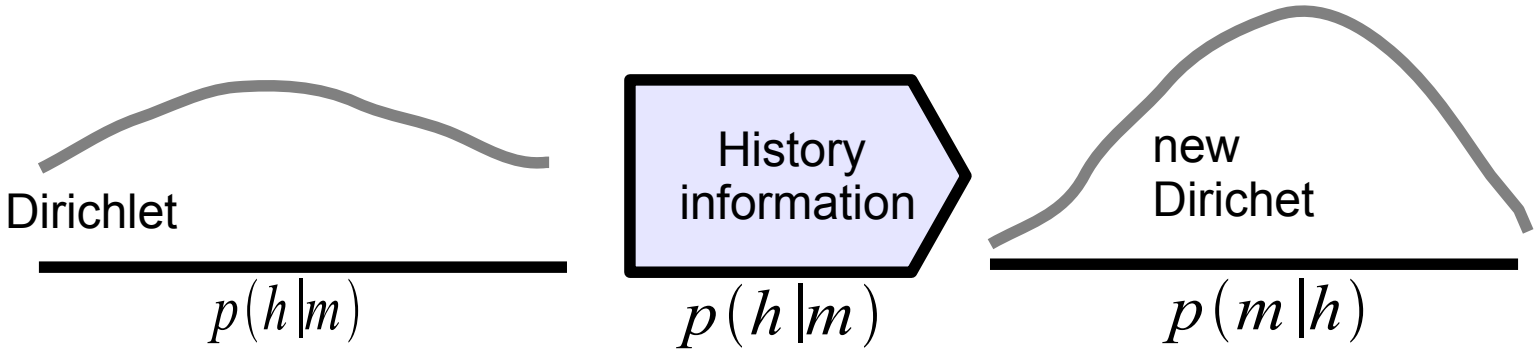
  - Error due to approximate POMDP solutions:

$$\epsilon_{pb} = 2\delta_b \frac{(R_{max} - R_{min})}{(1-\gamma)^2}$$

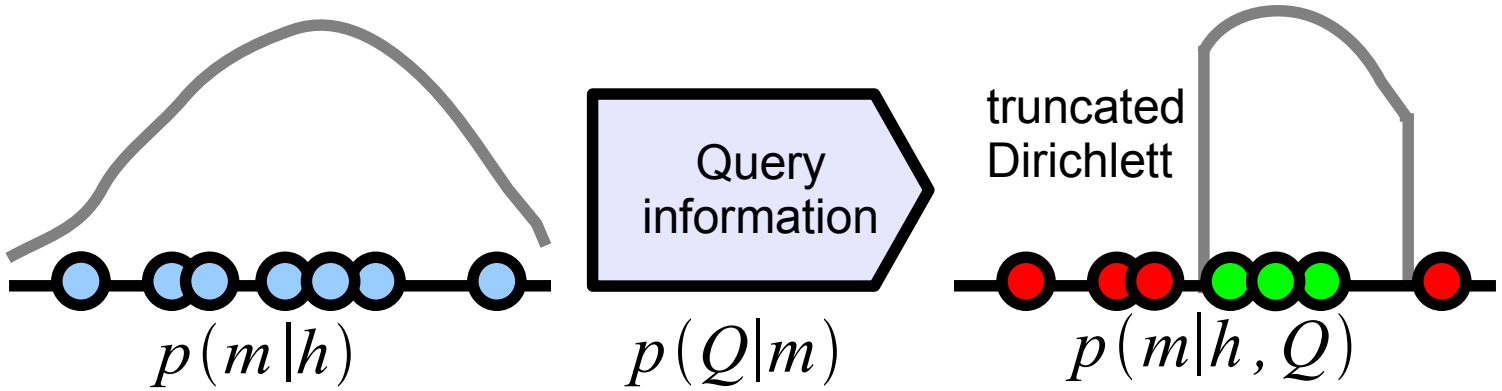
- Noting that $\zeta = \varepsilon_m + \varepsilon_{pb}$, set ε_m and ε_{pb} to trade between the number of belief samples and model samples.

Belief Update

- If we begin with a Dirichlet prior, the history information can be incorporated in closed form.



- We sample from the this updated distribution over models, throwing out samples that violate too many meta-queries.



Belief Update: History Information

- Use history information to analytically update Dirichlet prior over models:

$$p(m|h,Q) = \eta\, p(Q|m)\, p(h|m)\, p(m)$$

$p(m|h)$

- Dirichlet update requires state history; estimate the state sequence using the standard forward-backward algorithm.
- EM-like update; will converge to some local optimum.

Belief Update: Query information

- Next use importance sampling to sample POMDPs; choose a p(Q|m) that forgives noise in the policy query response and the approximate POMDP solution.

$$p(m|h,Q) = \eta\, p(Q|m)\, p(h|m)\, p(m)$$

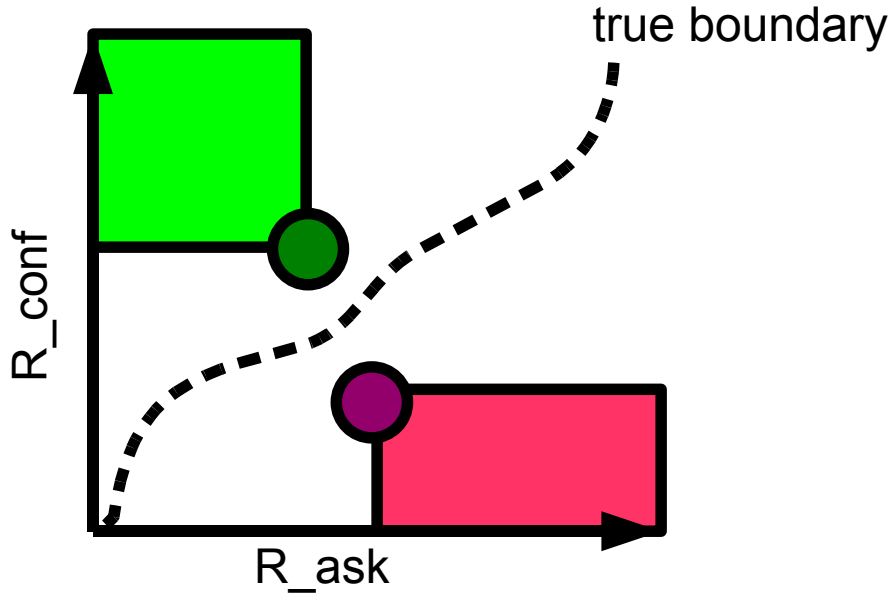
$$\frac{1}{1+k}\, u(k'-k)$$

- For a real time system, apply additional heuristics:
  - Try sampling until the original minimum error is reduced
  - Try a convex combination of new and good samples
  - Limit number of samples to try

CSAIL

Continuous Models: Belief Update

- If **only** the reward model is unknown, we can efficiently prune the reward space:



- We can use rejection sampling or MC MC techniques to sample from valid regions in the reward space.

Performance Guarantees

- We can provide a lower bound on the expected performance of our approach compared to the optimal policy:

$$V' > \eta\left(V - \frac{\xi}{(1-\gamma)}\right) + (1-\eta)\left(\frac{R_{min}}{(1-\gamma)}\right), \eta = \frac{(1-\gamma)(1-\delta)}{1-\gamma(1-\delta)}$$

- Since the Dirichlet counts are always increasing, we will eventually converge to some transition and observation model.

Termination Bounds

- To estimate if the probability of asking a meta-query after n more interactions is greater than ζ with confidence δ, we can:
  - Compute “worst posterier” by assigning interaction counts to make a flat Dirichlet posterior.
  - Sample POMDPs from the posterior.
  - Sample beliefs from the POMDPs.
  - Reject if $f(\zeta)$-proportion beliefs require meta-queries.
- We can set the number of POMDP, belief samples required, as well as $f(\zeta)$, based on our desired confidence.

Solving the POMDP Dialog Model

Value of a belief

Value of belief, action pair

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

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

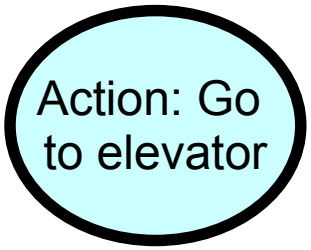
$$Q_n(b,a) = R(b,a) + \gamma \sum_{o \in O} O(o|b,a) V_{n-1}(b_a^o)$$

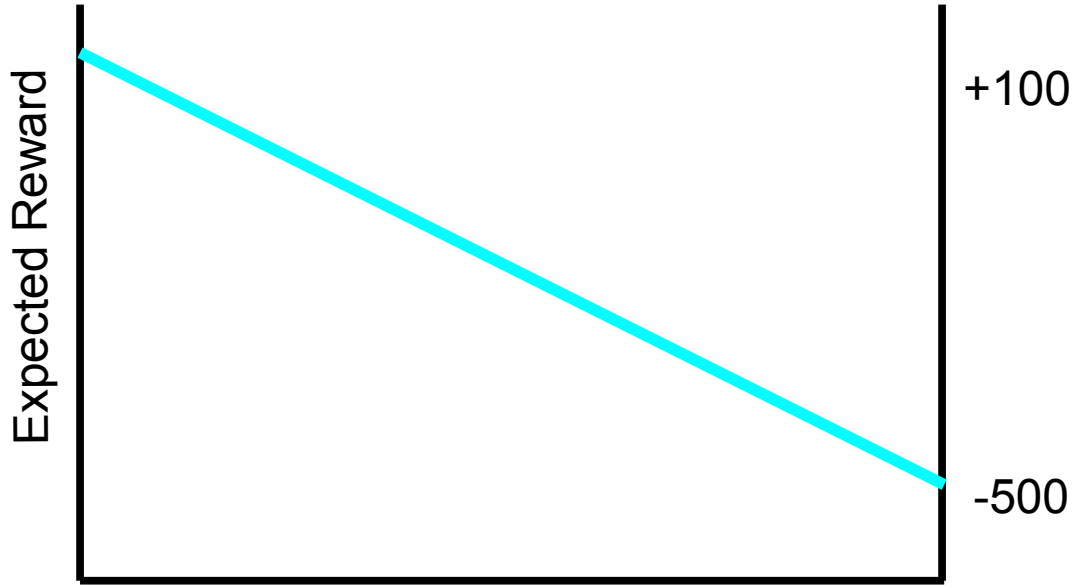
Current reward

Future Reward

Dialog Model: Solving the POMDP

We think of the previous recursions as building a policy tree...





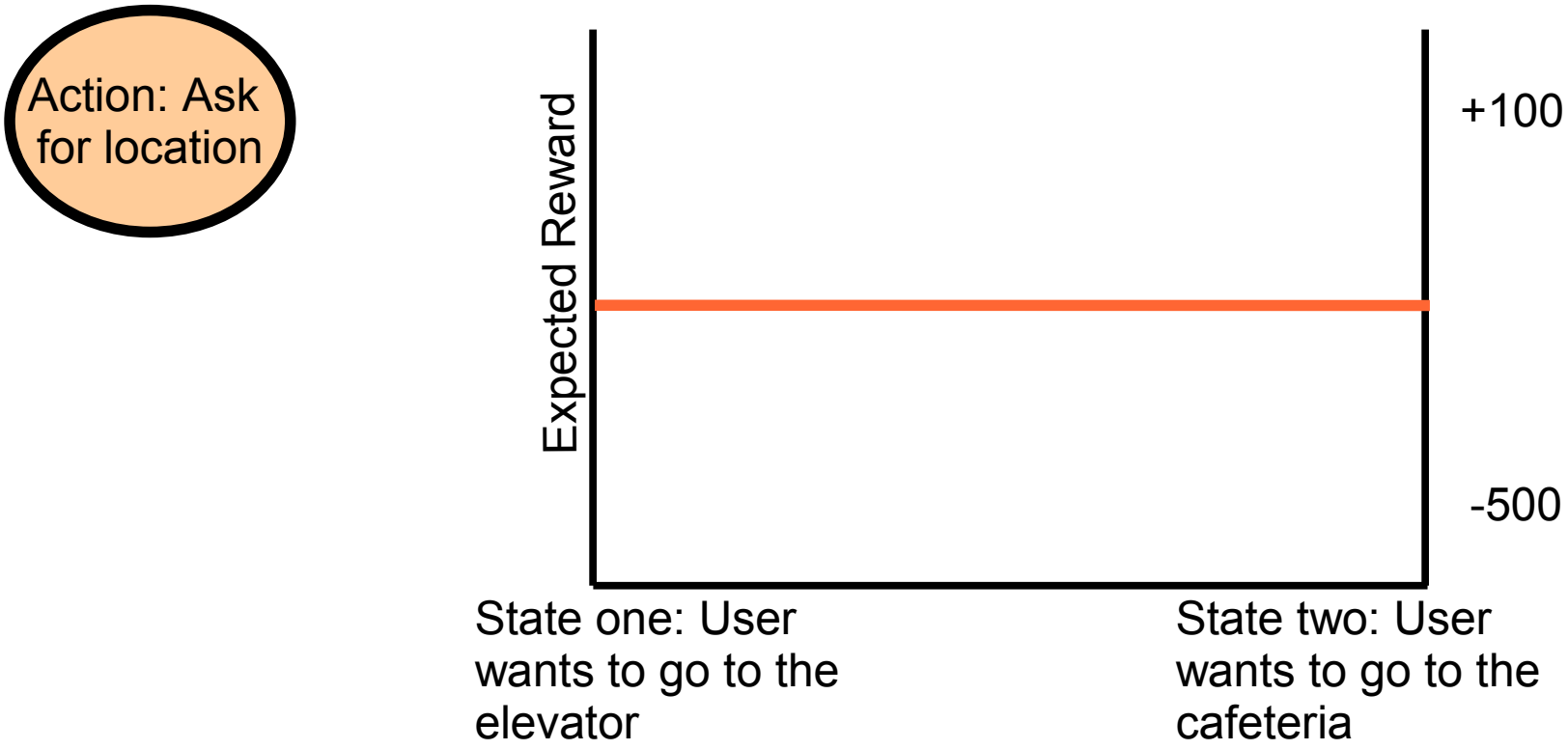
Certain that we're in state one:
User wants to go to the elevator

Certain in state two: User
wants to go to the cafeteria

CSAIL

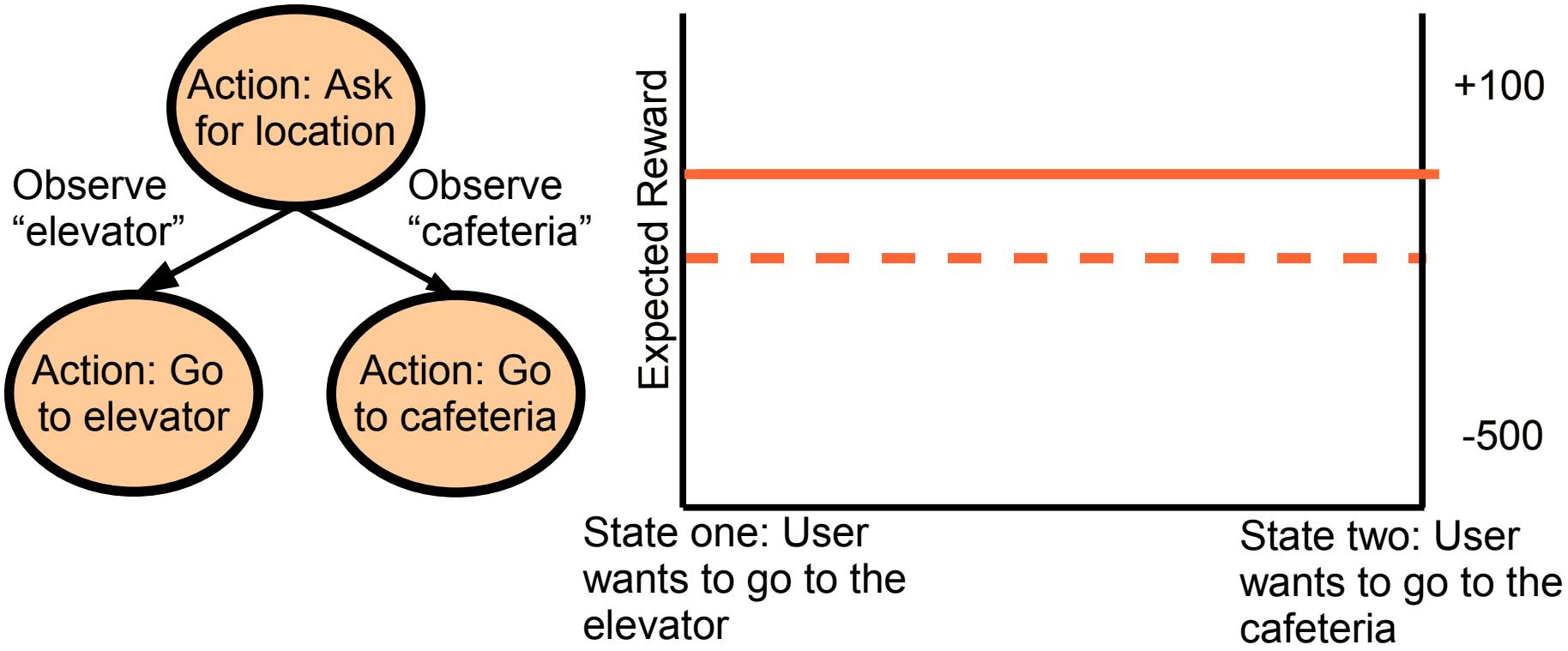
Dialog Model: Solving the POMDP

We think of the previous recursions as building a policy tree...



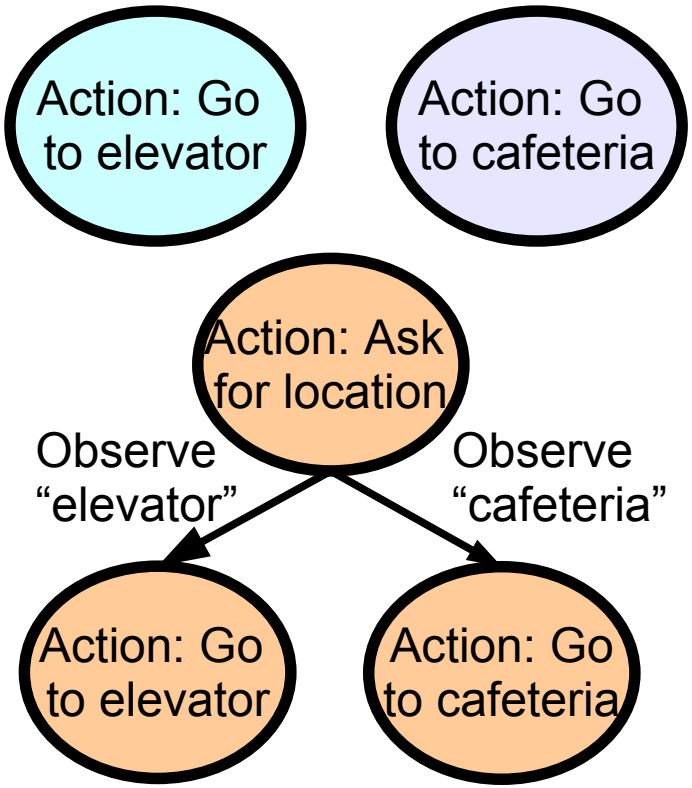
Dialog Model: Solving the POMDP

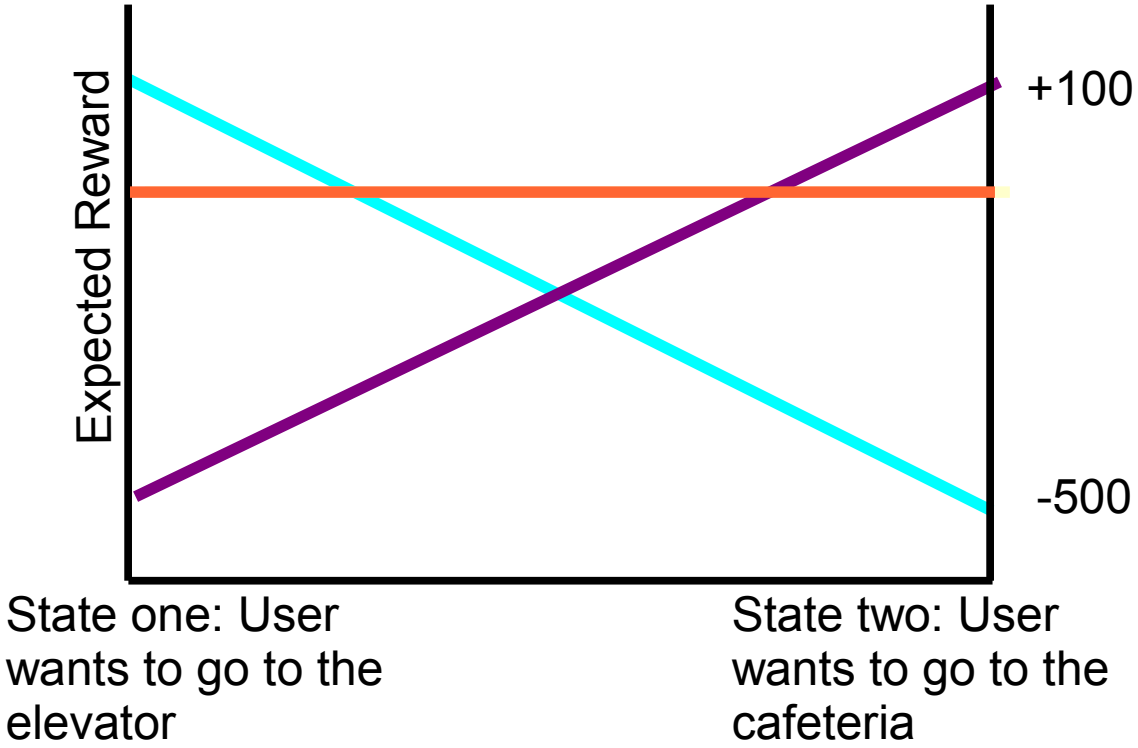
We think of the previous recursions as building a policy tree; planning ahead increases our expected reward.



Dialog Model: Solving the POMDP

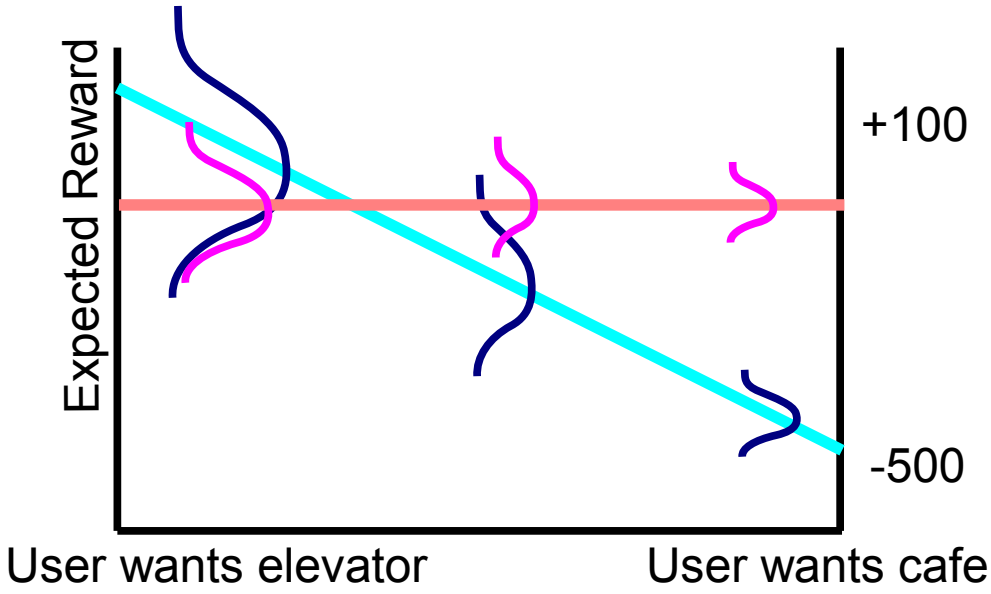
Given multiple trees, we can determine the most appropriate action:





1. Modeling Uncertainty: Placing Priors

- An expert provides guesses using “pre-observation counts.” For example: suppose I think P(error) = 10%
  - If I'm pretty certain, guess “saw 100 errors in 1000 tests”
  - If I'm not sure, might guess “saw 1 error in 10 tests”
- Parameter uncertainty induces uncertainty in the value function.

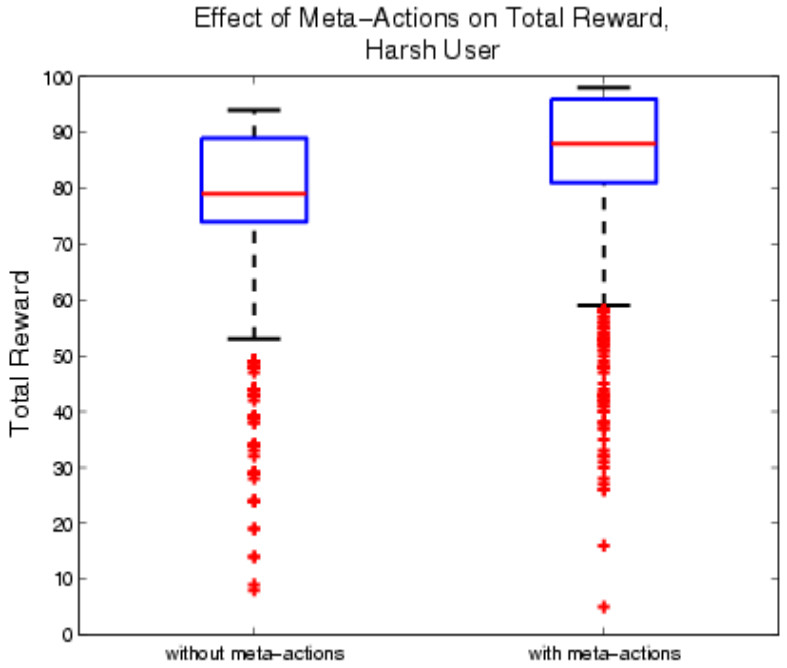


Prior Work

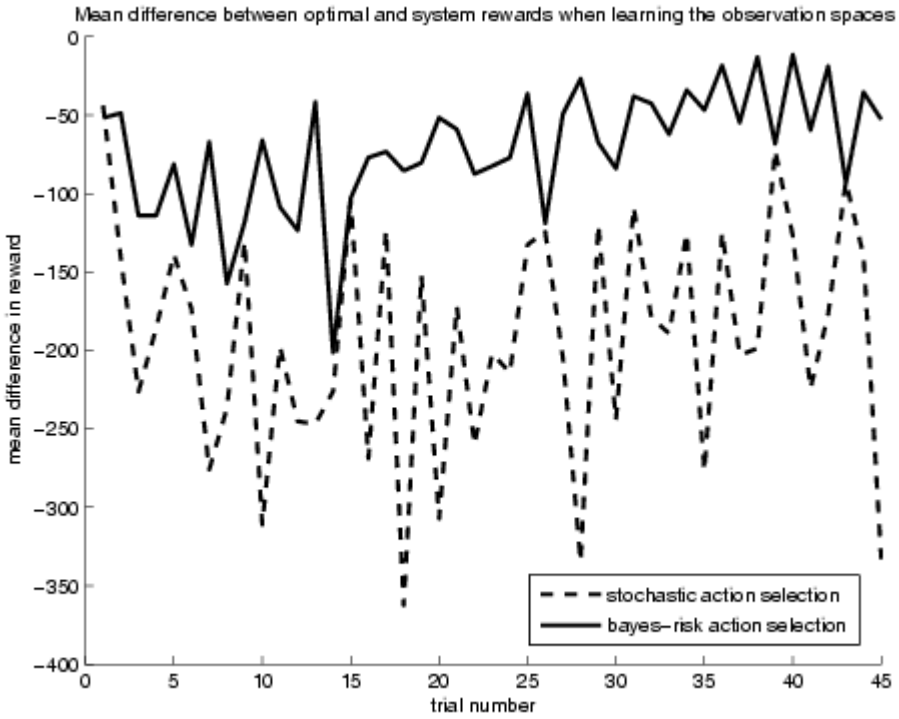
- POMDP Dialog Management: (assume model known)
  - Roy et. al.: nursing home aide
  - Williams and Young: automated telephone operator
- Bayesian (PO)MDP Model Learning:
  - Dearden et. al.: Bayesian MDP model learning
  - Beetle (Poupart et. al.): frame unknown MDP as a continuous state POMDP
  - Medusa (Jaulmes et. al.): sample from a distribution over POMDPs; use the sample for action selection

Meta-Queries (Discrete Models)

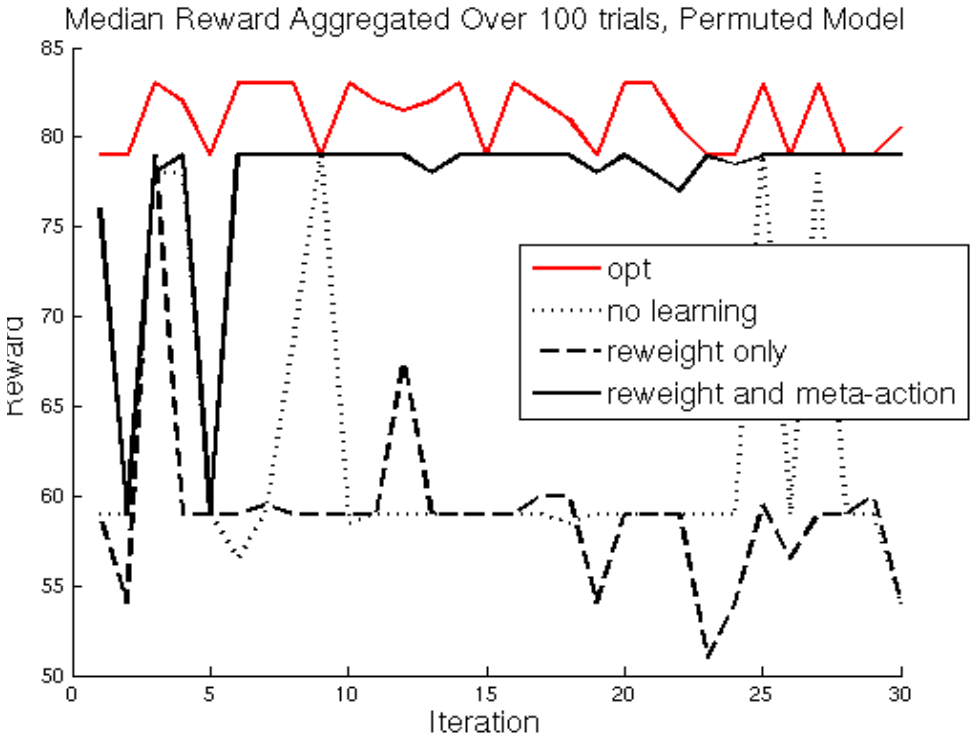
- Choose sets of parameters that produce different policies; let each of these be a user preference model.
- Design meta-action queries to differentiate between the models.
- Solve just like the parameter POMDP.



BR Action Selection (Continuous Models)



Observation Learning



Reward Learning

Dialogs with Meta-Queries

- Meta-queries decrease with experience (from user trials, significant at the 5% level)

Times a place requested	First	Second	Third
Total number of requests	29	15	8
Requests with meta-queries	22	11	2
Proportion with meta-queries	0.76	0.73	0.25