Reinforcement Learning with Limited Reinforcement

Using Bayes Risk for Active Learning in POMDPs

Finale Doshi
Nick Roy
Joelle Pineau

CSAIL

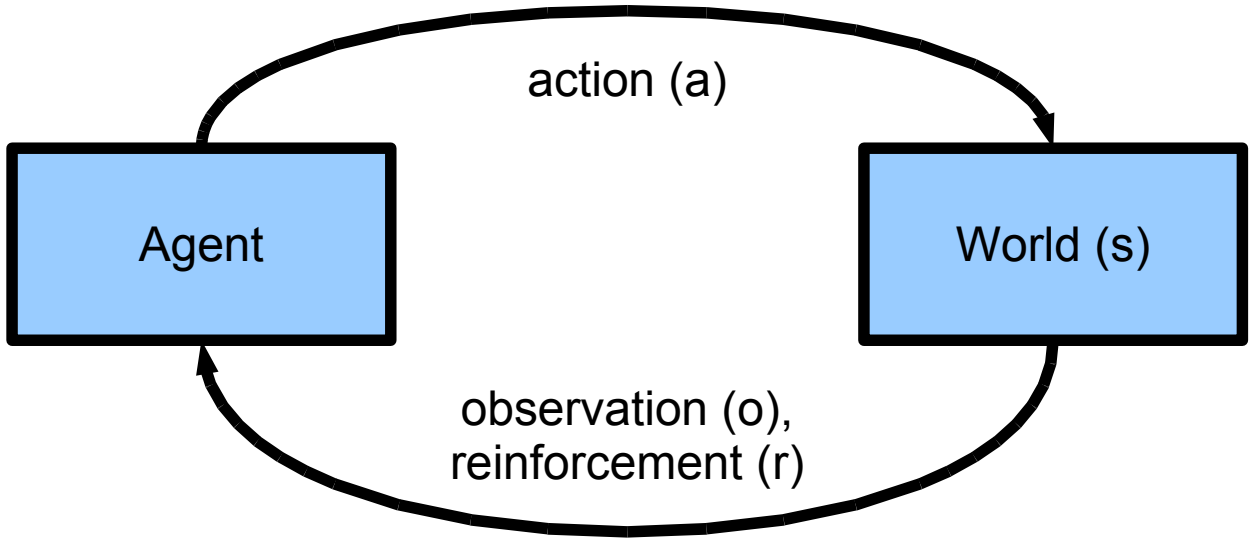
Outline and Motivation

How can an agent robustly learn to act in an unknown environment?

- Traditional Reinforcement Learning
  - Background and drawbacks
- Our Approach: Active Learning using Bayes Risk
  - Algorithm overview
  - Performance properties
- Results

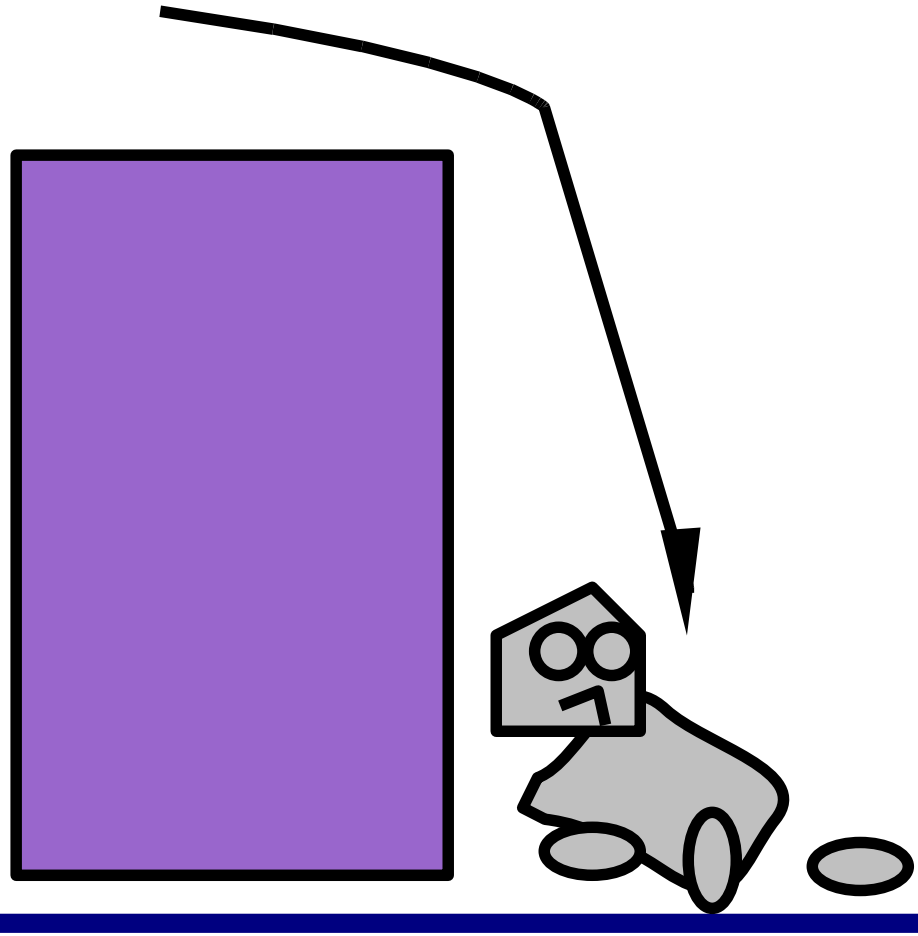
Reinforcement Learning Paradigm

- Agent performs actions, receives observations and rewards.
- Assume a Markov world; world dynamics consists of reward R(s,a), transition T(s'|s,a), and observation O(o|s,a) models.
- Agent must trade between learning about the world (exploration) and getting rewards (exploitation).



Problems with the Traditional Framework

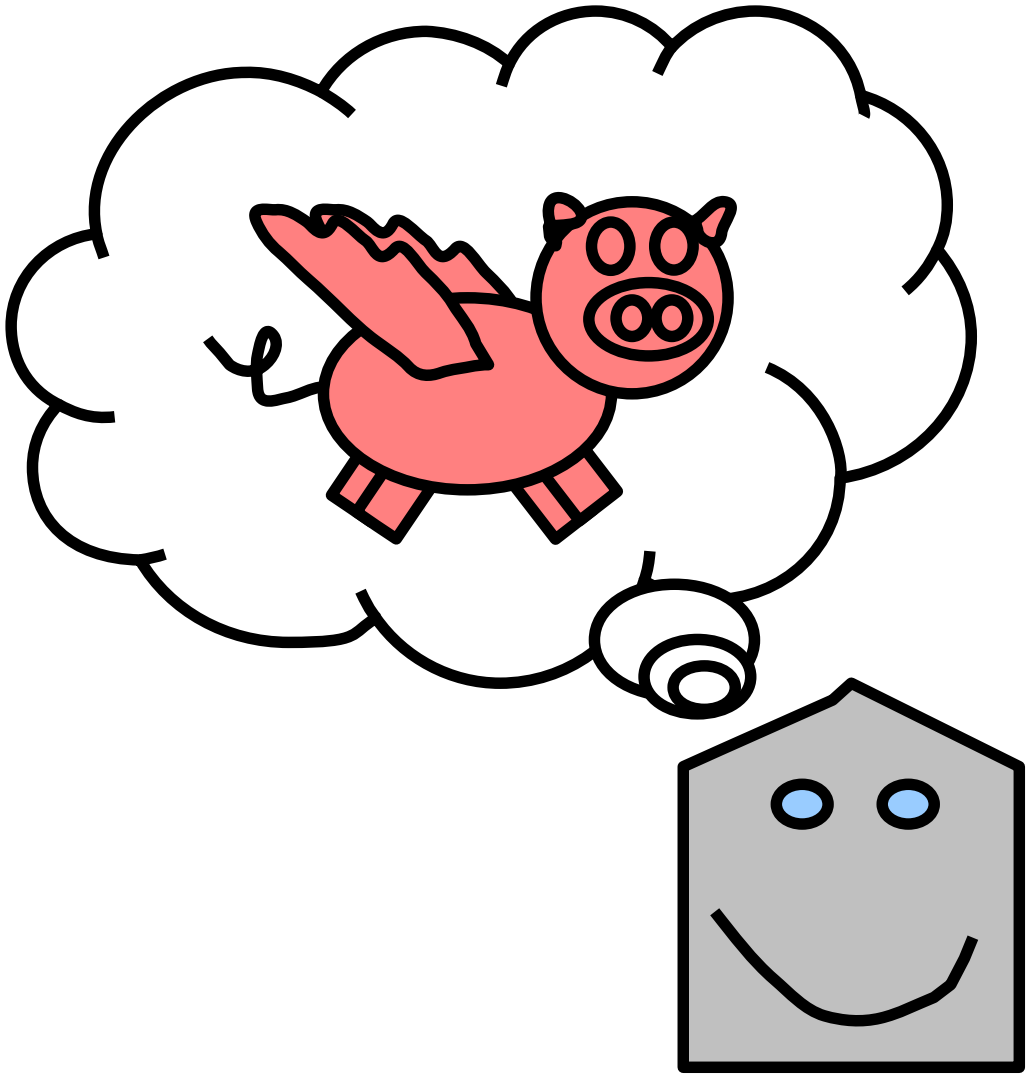
- Must make mistakes to learn; can be undesirable (ex. robotic wheelchair)



CSAIL

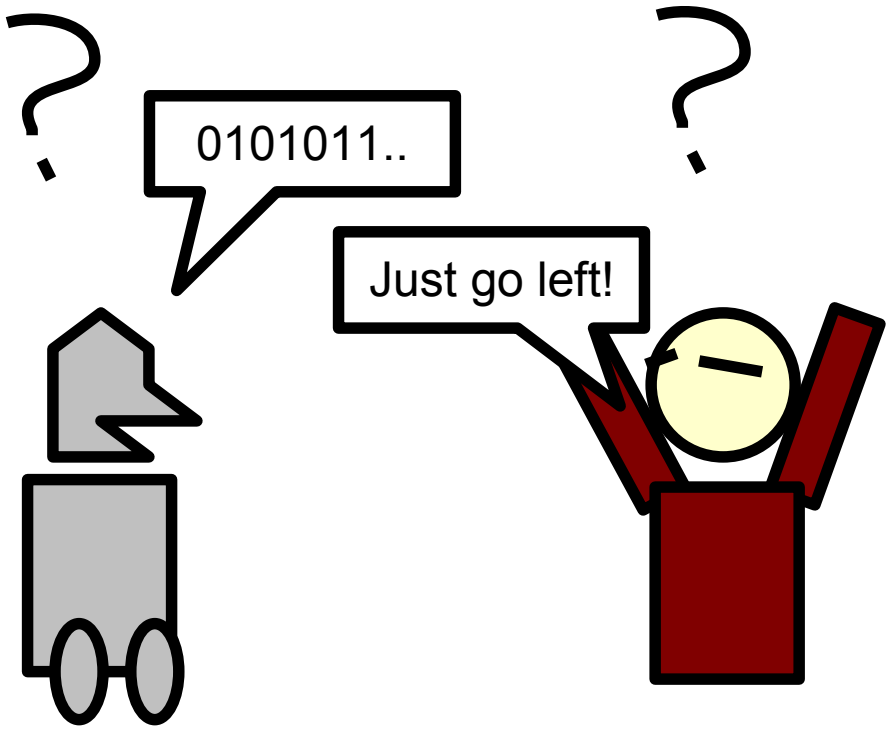
Problems with the Traditional Framework

- Must make mistakes to learn; can be undesirable (ex. robotic wheelchair).
- Aside from convergence and coverage issues, no reasoning about the uncertainty in the policy due to partial information.



Problems with the Traditional Framework

- Must make mistakes to learn; can be undesirable (ex. robotic wheelchair).
- Aside from convergence and coverage issues, no reasoning about the uncertainty in the policy due to partial information.
- Traditional reinforcement required; can't use policy info or more natural ways to ask for help (ex. may be useful in wheelchair dialog scenario).

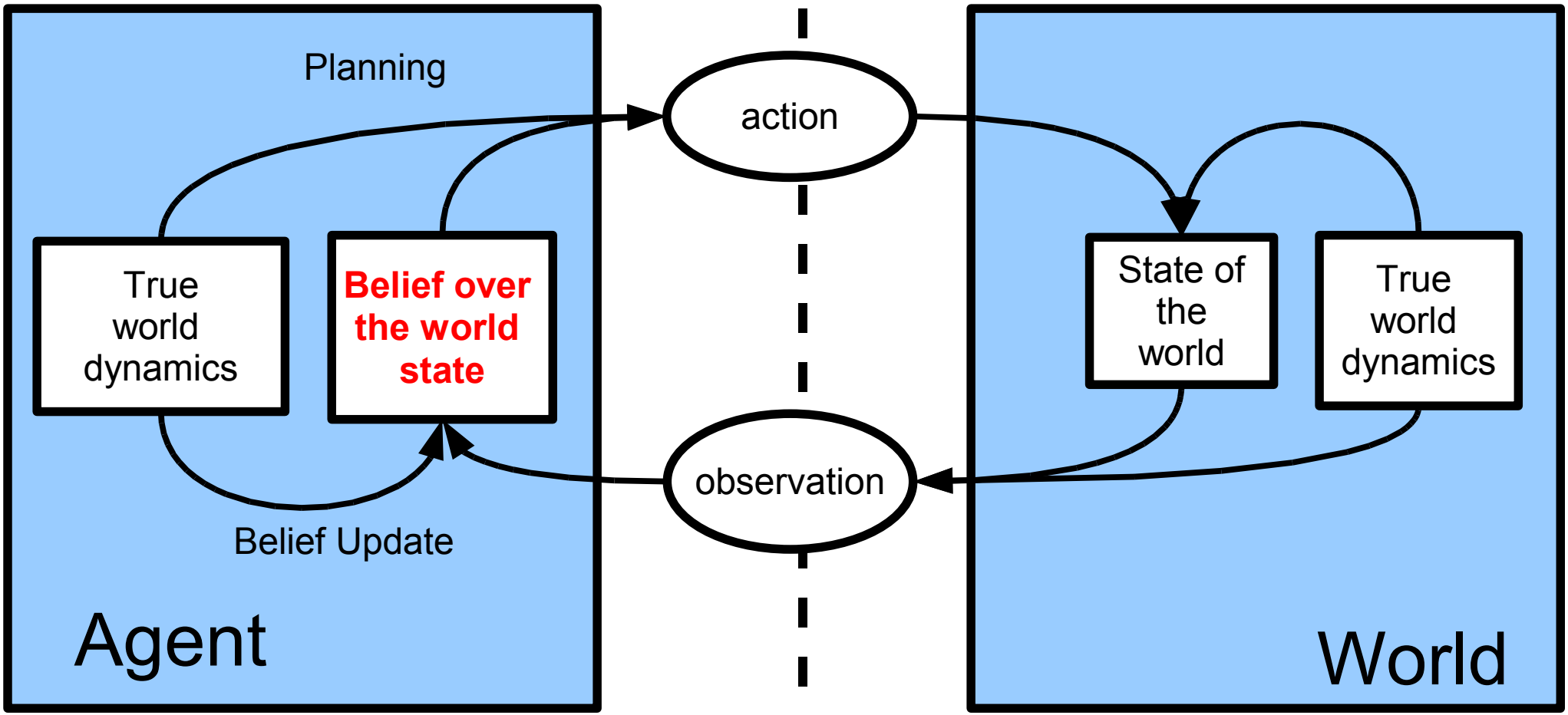


Our Approach

- **POMDP framework:** a Bayesian approach to model uncertainty (solved approximately using Bayes risk).
  - The agent is **aware of its model uncertainty**.
- **Meta-queries:** Ask for policy information when confused.
  - The agent can **actively reduce model uncertainty.**
- Resulting approach **combines Bayesian and inverse reinforcement learning** to robustly learn the reward, transition, and observation models simultaneously.

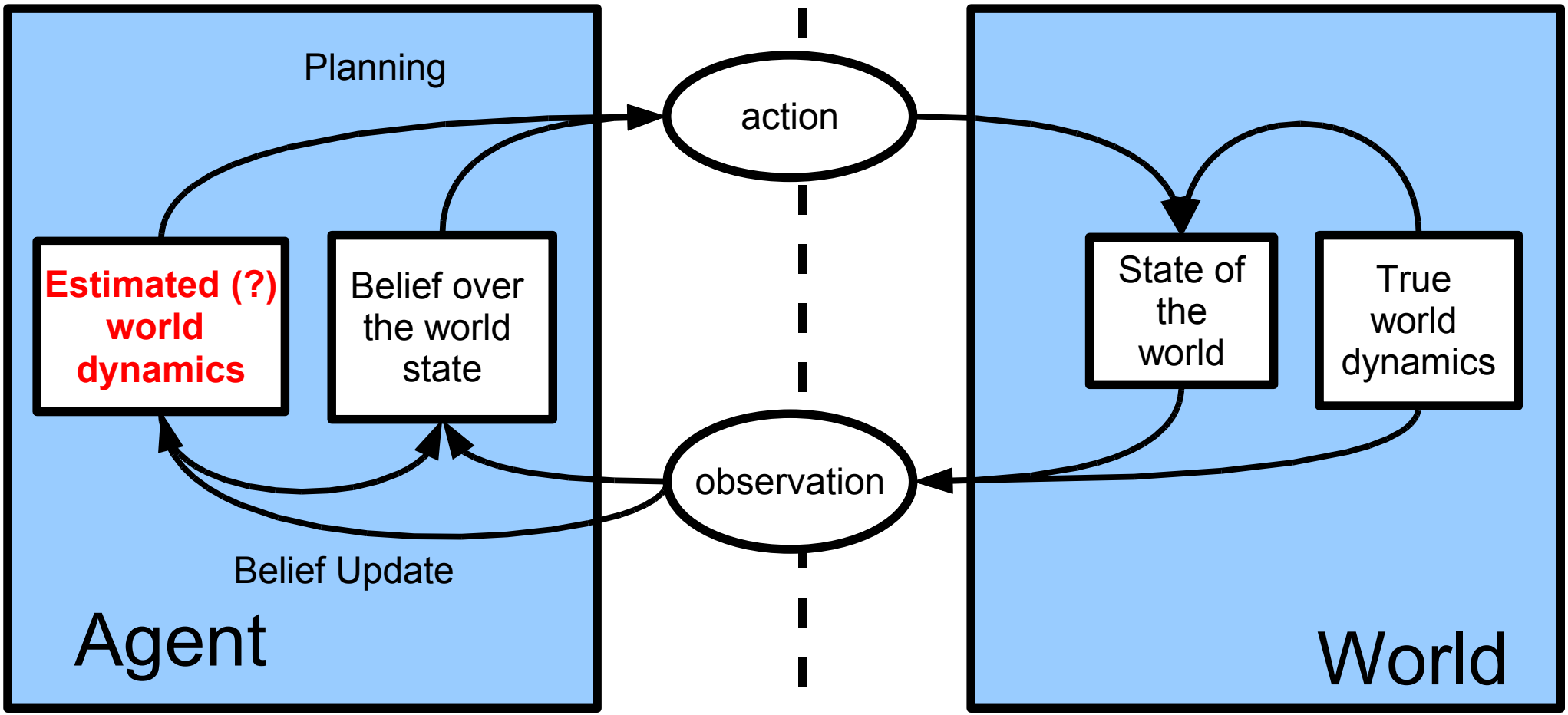
The POMDP Planning Process

When solved, the POMDP optimally trades between gathering information about the state and gathering reward.



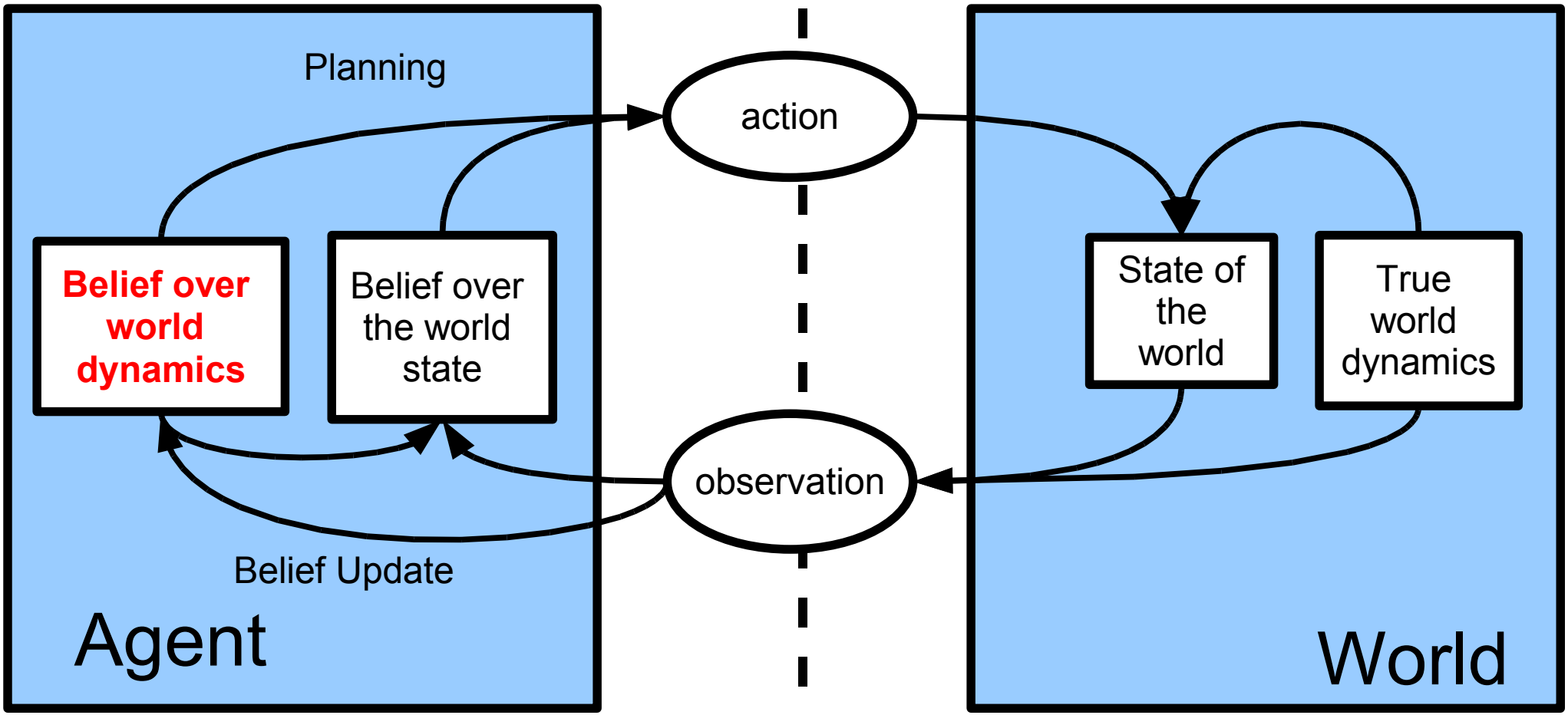
Planning with Uncertain Models

However, models are hard to come by! One option is to keep an estimate of the true model.



Planning with Uncertain Models

But, if we think of the model as hidden state, dealing with model uncertainty is equivalent to dealing with state uncertainty.

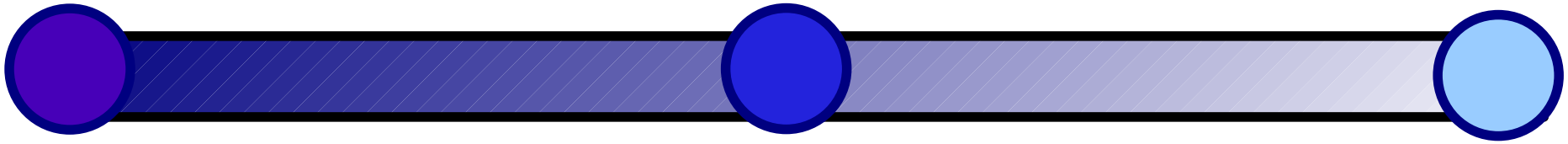


Planning with Uncertain Models

Ignore uncertainty:
fast, not robust

Plan with parameters as hidden state:
robust but slow

Planning with Uncertain Models

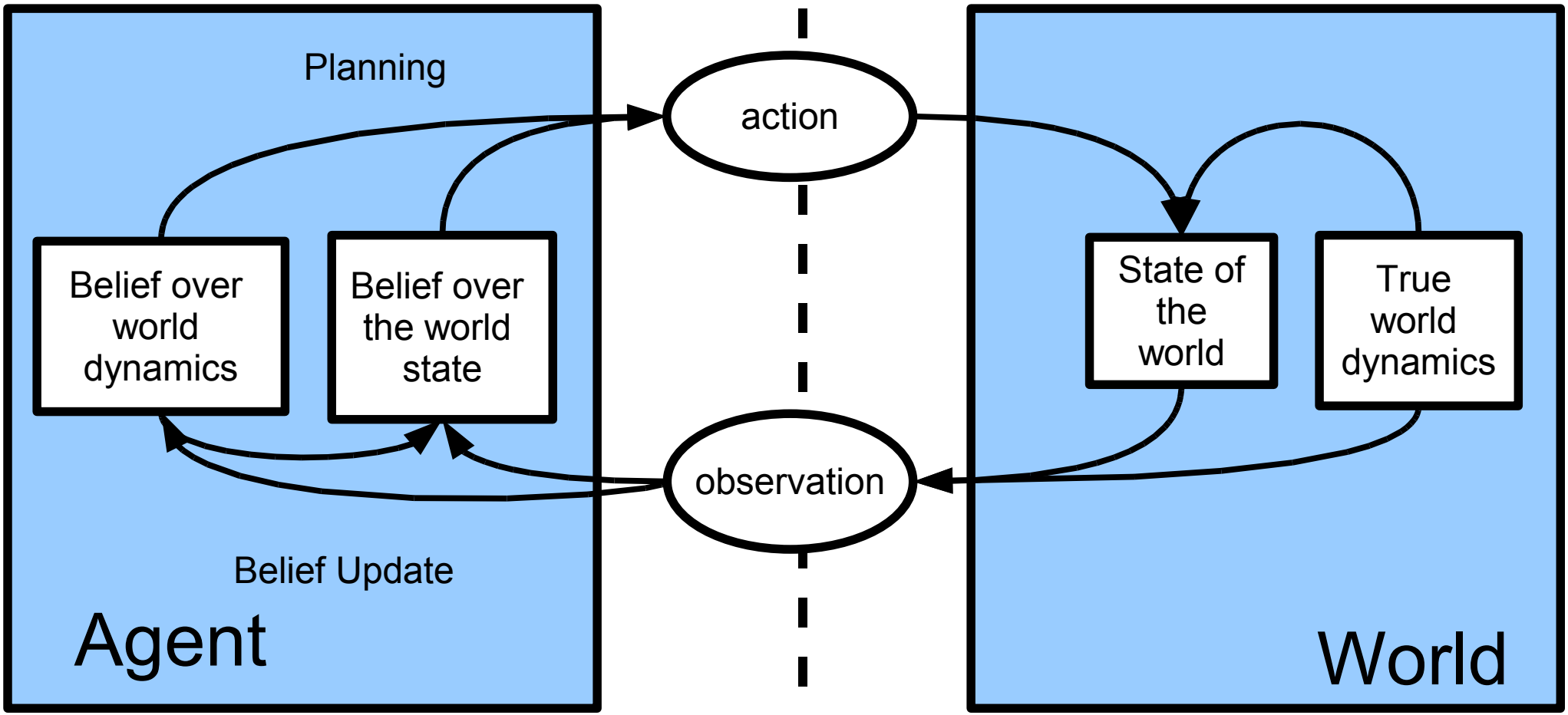


Ignore uncertainty:
fast, not robust

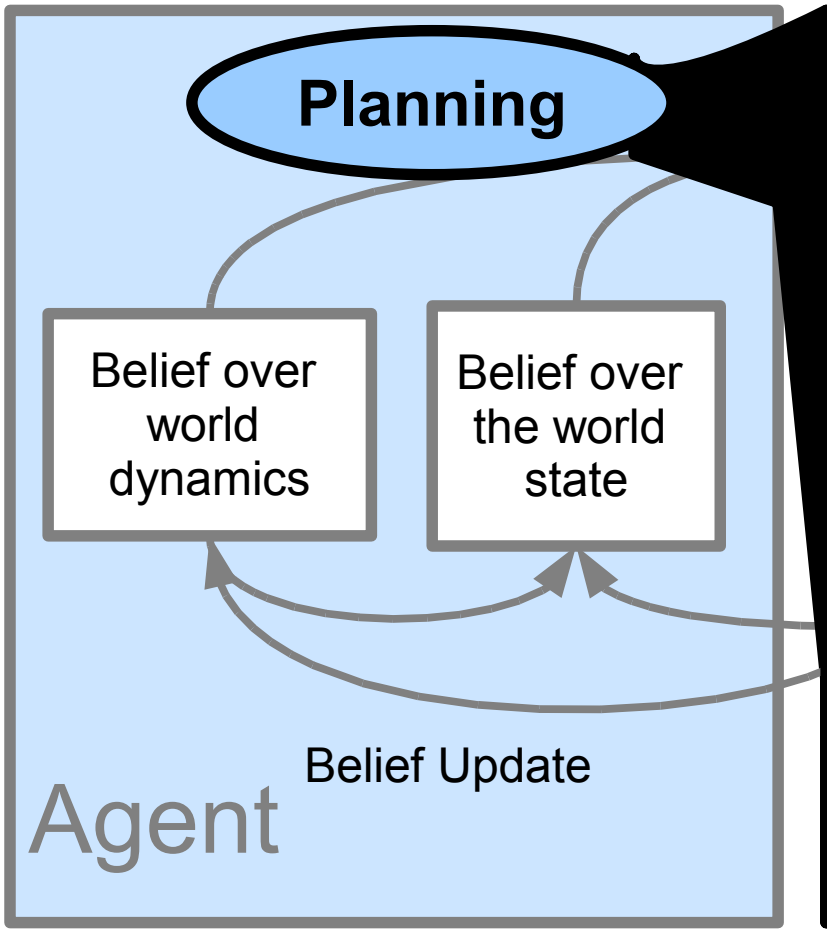
Approximate planning with Bayes risk, meta-queries

Plan with parameters as hidden state:
robust but slow

The Model-Uncertainty POMDP



Action Selection



- Use Bayes Risk to find the safest action.
- If this action is too risky, ask for help.

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 meta-query cost, ask for help.

- Evaluate the Bayes Risk integral approximately using sampled POMDPs:

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

(We can bound the approximation error.)

CSAIL

Meta-Queries

Idea: ask for policy help by giving a human aide a sense of the agent's uncertainty.

Benefits:

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

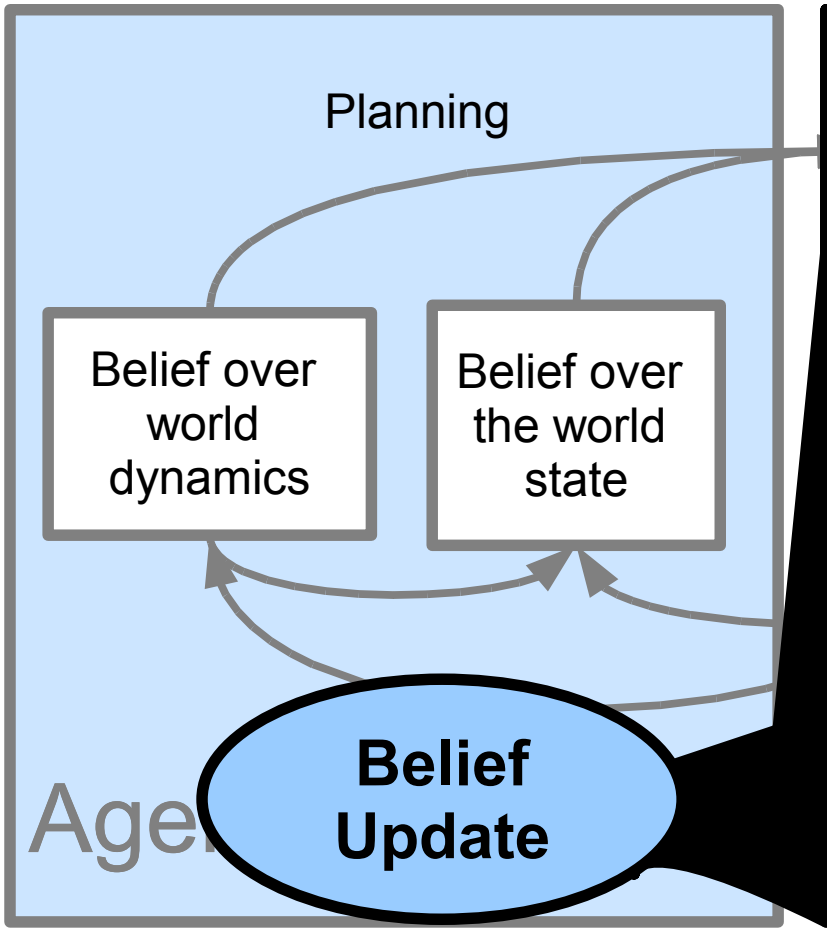
Meta-Queries

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 to determine the correct action; thus a **meta-query results in discovering the optimal action**.

Belief Update

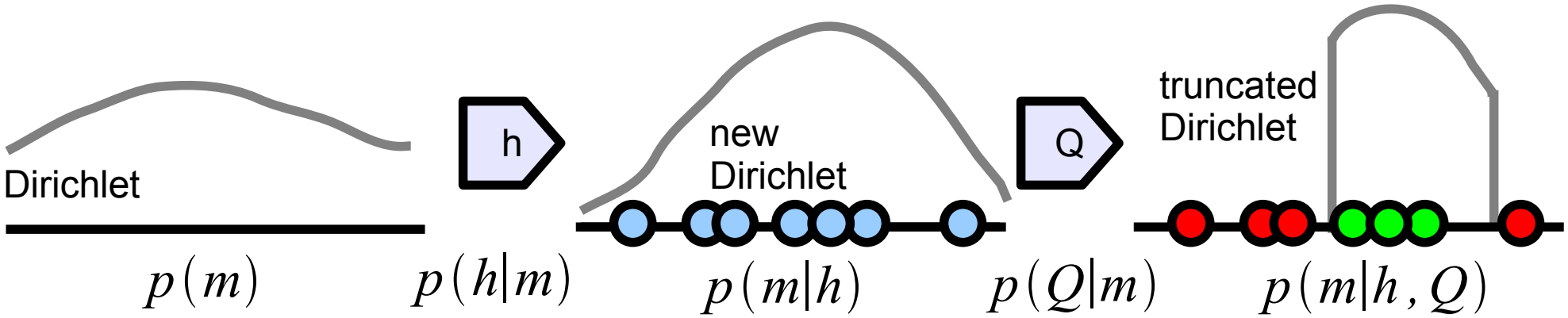


We need to incorporate two sources of information:

- h: Most recent history of actions and observations
- q: Set of (query, response) pairs that we have asked

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.)

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.

CSAIL

Belief Update: Query information

- Next use rejection sampling to reject POMDPs that do not agree with the query set.

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

$\delta(k)$

- For a real time system, apply additional heuristics:
  - Use p(Q|m) = $\frac{1}{1+k} u(k'-k)$
  - Only sample until the original minimum error is reduced
  - Limit number of samples to try

Performance Guarantees

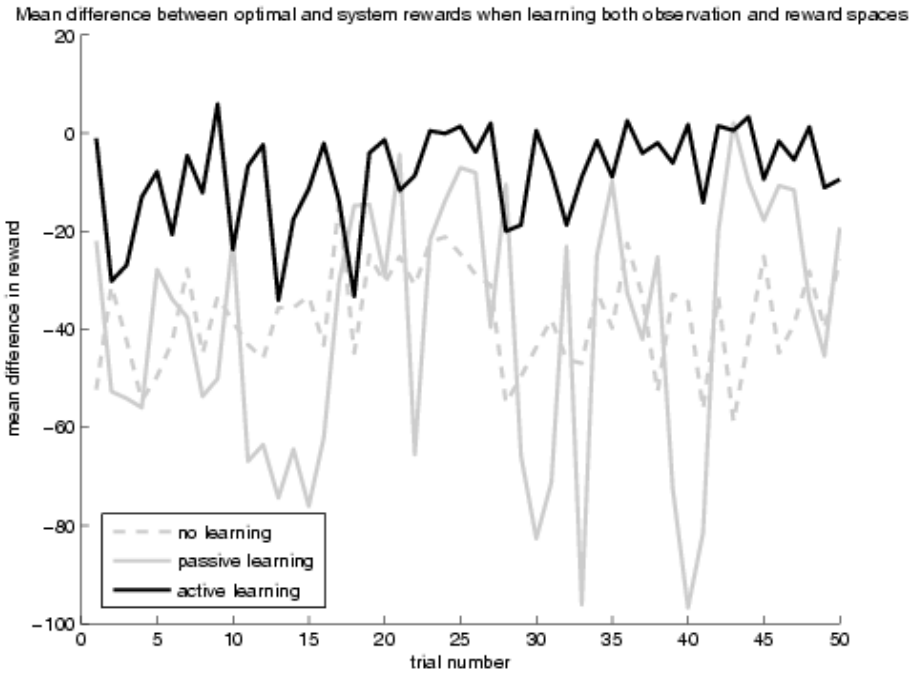
- We can provide a lower bound on the expected performance of our approach compared to the optimal policy:

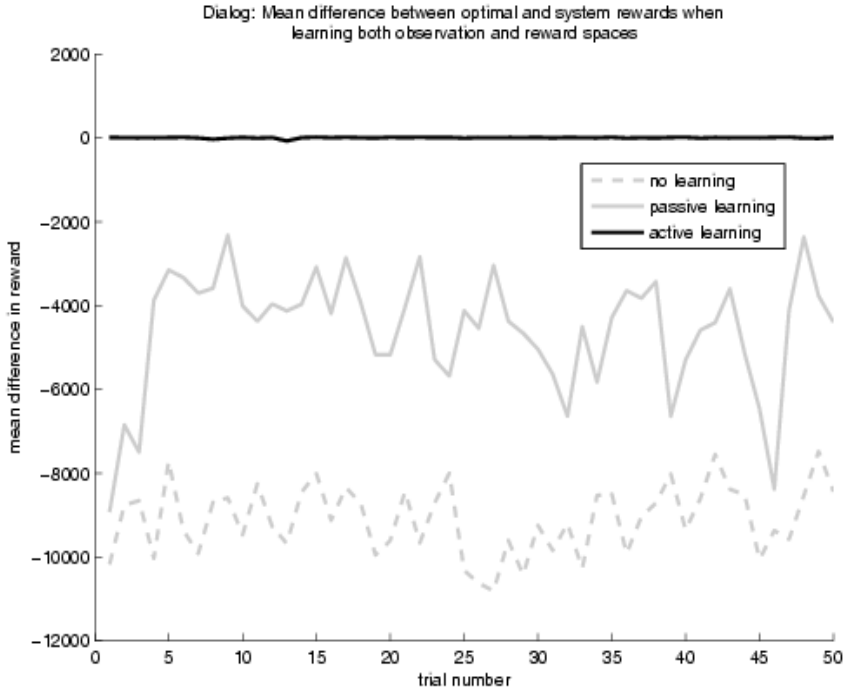
$$V' > \eta\left(V - \frac{\xi}{(1-\gamma)}\right) + (1-\eta)\left(\frac{R_{min}}{(1-\gamma)}\right), \eta = \frac{(1-\gamma)(1-\delta)}{1-\gamma(1-\delta)}$$

- We will eventually converge to a transition, observation, and reward model.

Results

Simulation Results: Full Model Learning





Reasonable Prior

Non-informative Prior

(similar results on various standard POMDP models, including hallway and tigergrid)

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

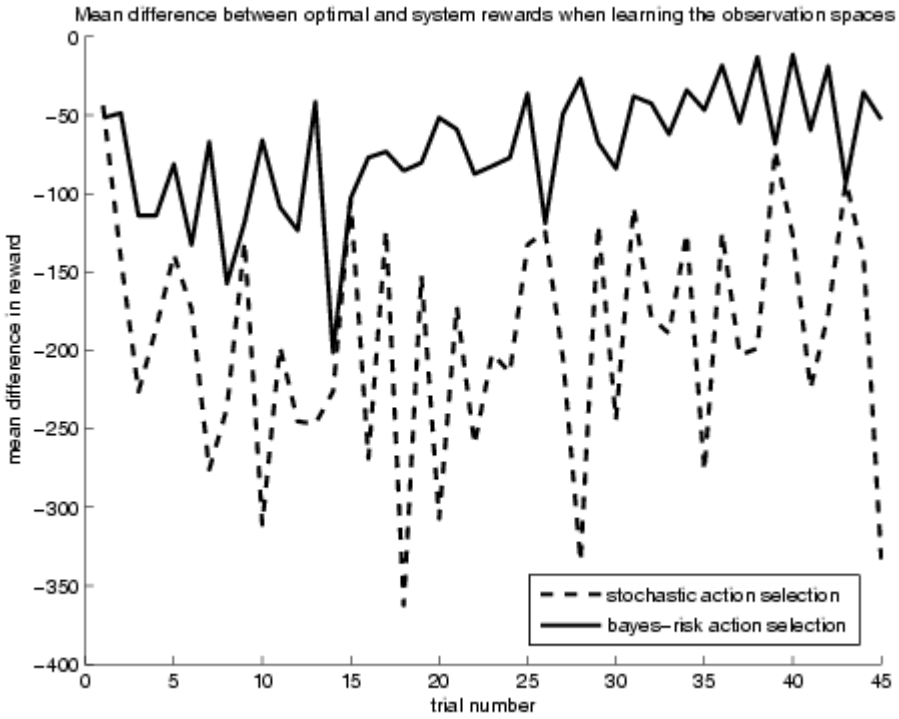
Later Conversation:
User: What's the forecast for today?
Robot: Do you want the weather?
User: Yup.
Robot: Showers

Conclusions and Future Work

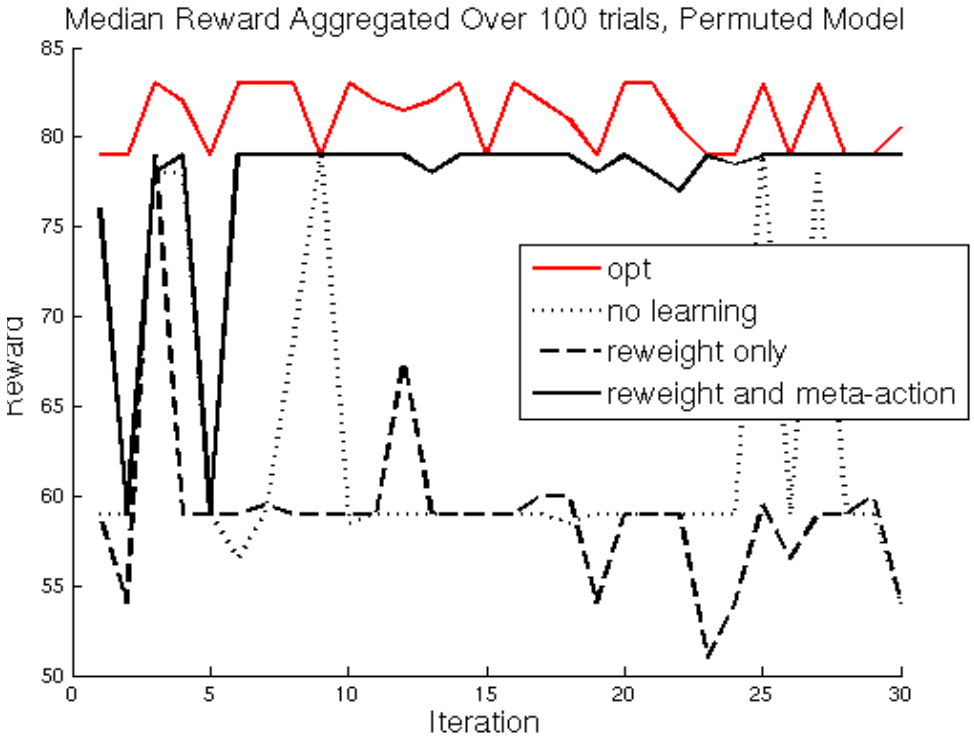
- Although POMDP models require many parameters, we can learn those parameters online.
  - Bayes risk action selection lets the agent act robustly in the face of model uncertainty
  - The learning process can be further improved by incorporating meta-queries.
- Extensions
  - Improve POMDP sampling techniques (or find a closed-form approximation to the POMDP posterior).
  - Approaches to finding policy information from humans (improved meta-queries).

Thank-you!

BR Action Selection (Continuous Models)



Observation Learning



Reward Learning

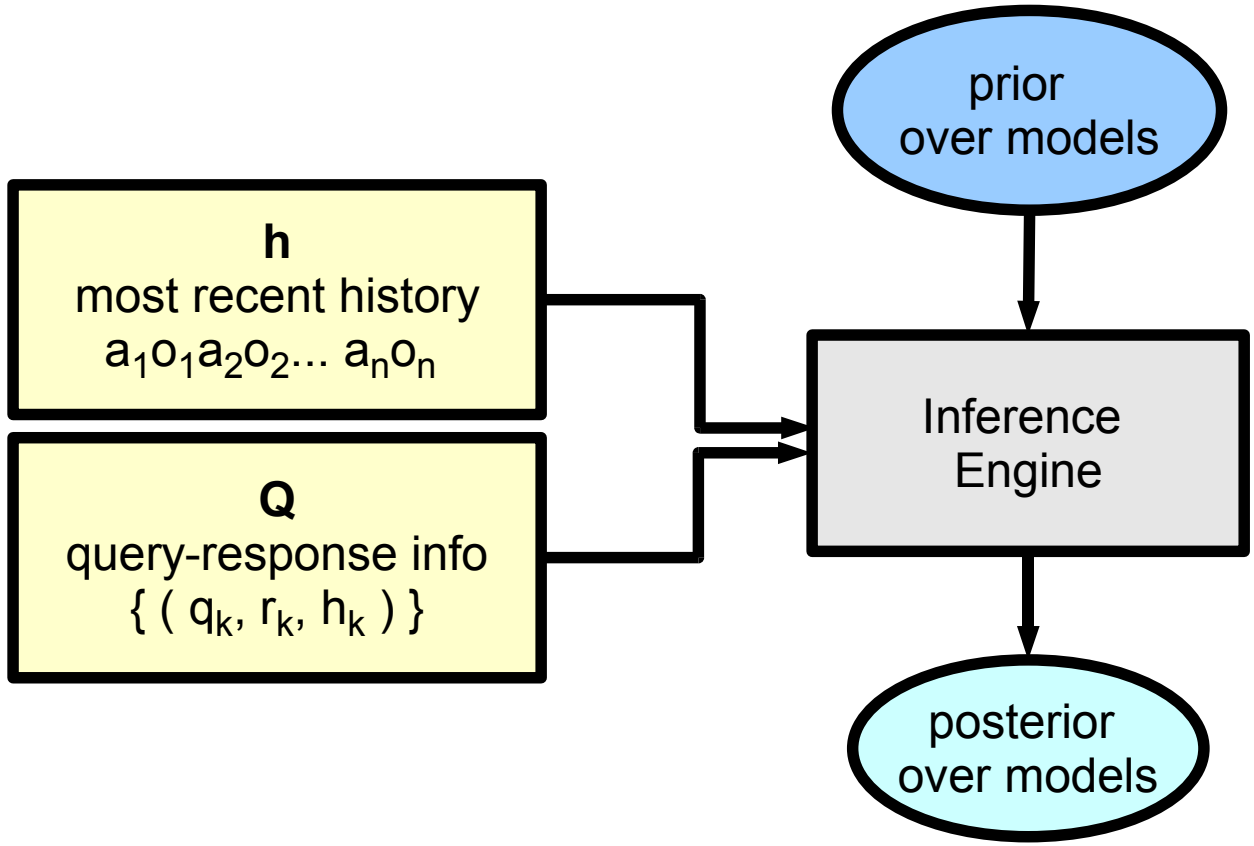
User Test Results: Example Dialog 2

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.

Belief Update

We have two sources of information to update our prior over models.



Related Work in Bayesian 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

Solving a known POMDP 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

CSAIL

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:

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

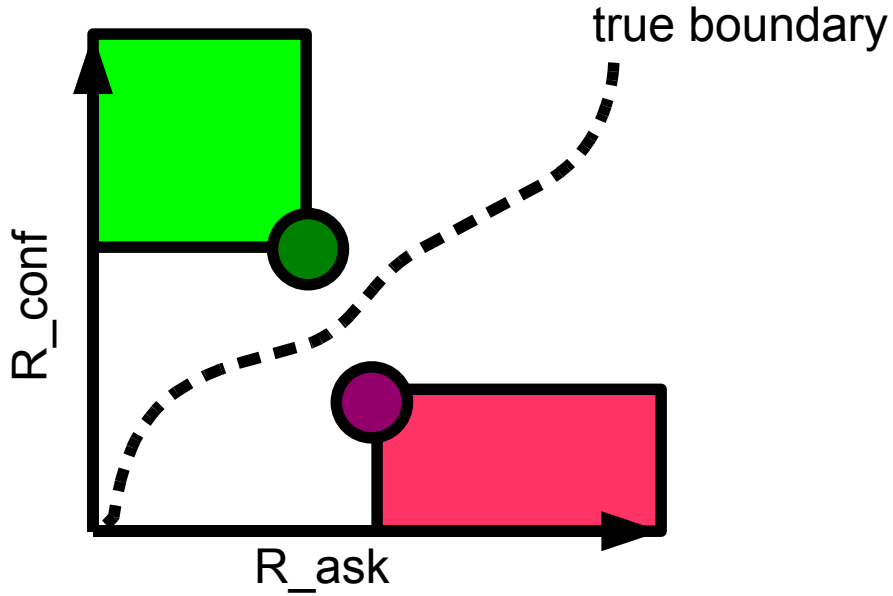
  - Error due to approximate POMDP solutions:

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

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

Continuous Models: Belief Update

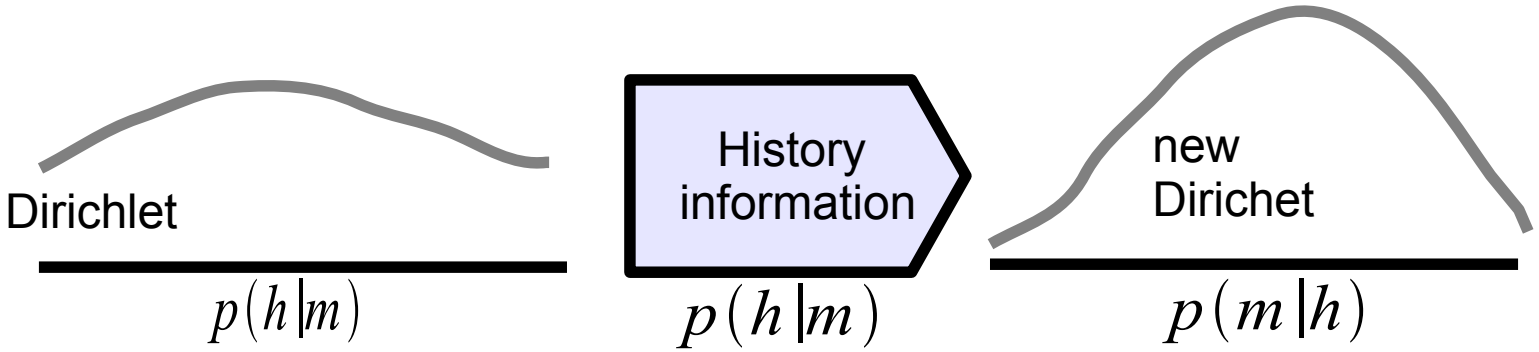
- If **only** the reward model is unknown, we can efficiently prune the reward space:



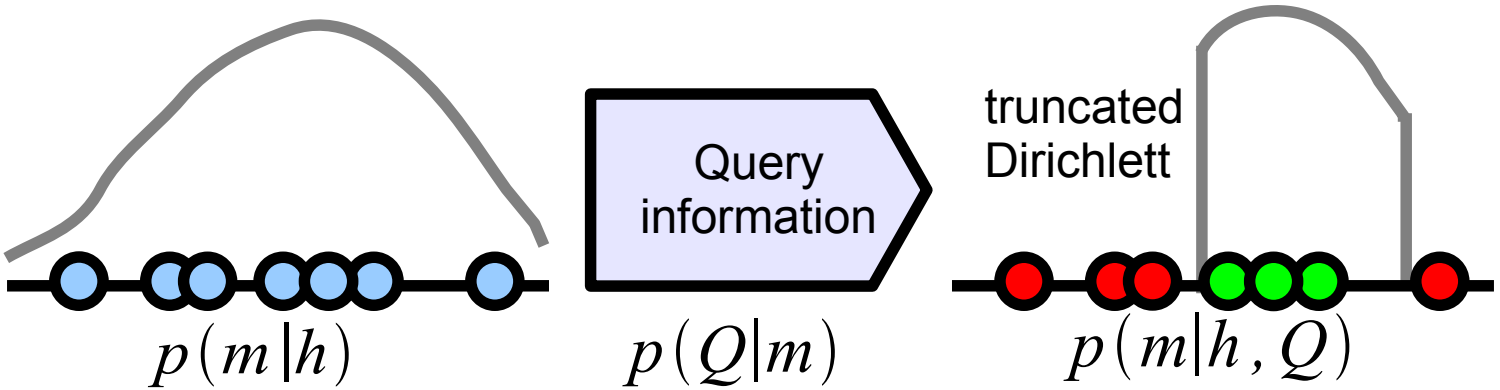
- We can use rejection sampling or MC MC techniques to sample from valid regions in the reward space.

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.



Termination Procedure

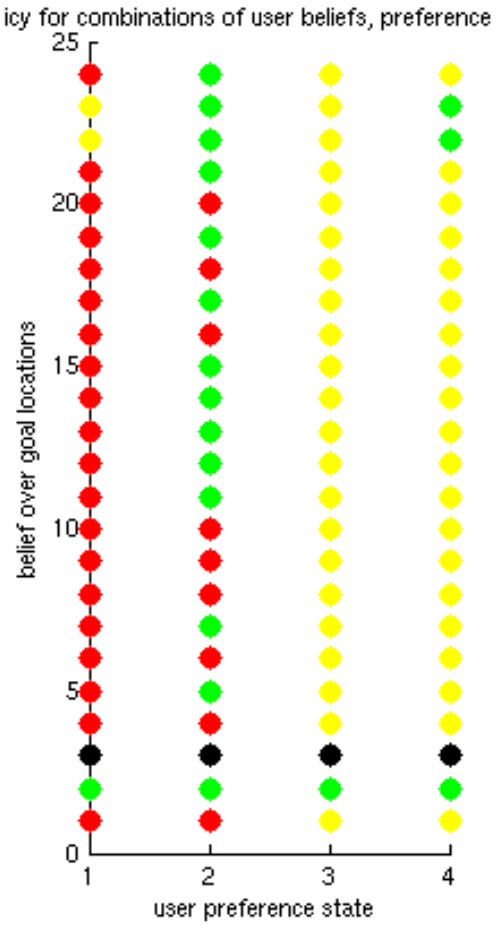
- To estimate if the probability of asking a meta-query after n more interactions is greater than ζ with confidence δ, we can:
  - Compute “worst posterier” by assigning interaction counts to make a flat Dirichlet posterior.
  - Sample POMDPs from the posterior.
  - Sample beliefs from the POMDPs.
  - 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.

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.



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.